UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

FACULTAD DE INGENIERIA

Selección de modelos de ritmo para la simulación de líneas melódicas[1]

Verónica Alejandra Rumbo Martínez

Programa de Posgrado en Ingeniería Matemática
Facultad de Ingeniería
Universidad de la República

Montevideo – Uruguay

Abril de 2019

[1]La investigación que da origen a los resultados presentados en la siguiente publicación recibió fondos de la Agencia Nacional de Investigación e Innovación bajo el código POS_NAC_2016_1_130953.

UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

FACULTAD DE INGENIERIA

Selección de modelos de ritmo para la simulación de líneas melódicas[1]

Verónica Alejandra Rumbo Martínez

Tesis de Maestría presentada al Programa de Posgrado en Ingeniería Matemática, Facultad de Ingeniería de la Universidad de la República, como parte de los requisitos necesarios para la obtención del título de Magíster en Ingeniería Matemática.

Directores:
Prof. Dr. Ernesto Mordecki Pupko
Prof. Dr. Martín Rocamora Martínez

Director académico:
Prof. Dr. Ernesto Mordecki Pupko

Montevideo – Uruguay

Abril de 2019

[1]La investigación que da origen a los resultados presentados en la siguiente publicación recibió fondos de la Agencia Nacional de Investigación e Innovación bajo el código POS_NAC_2016_1_130953.

Rumbo Martínez, Verónica Alejandra

Selección de modelos de ritmo para la simulación de líneas melódicas[a] / Verónica Alejandra Rumbo Martínez. - Montevideo: Universidad de la República, Facultad de Ingeniería, 2019.

VIII, 78 p. 29, 7cm.

Directores:

Ernesto Mordecki Pupko

Martín Rocamora Martínez

Director académico:

Ernesto Mordecki Pupko

Tesis de Maestría – Universidad de la República, Programa en Ingeniería Matemática, 2019.

Referencias bibliográficas: p. 68 – 70.

1. Probabilidad, 2. Estadística, 3. Música, 4. Cadenas de Markov, 5. MDL, 6. Simulación. I. Mordecki Pupko, Ernesto, Rocamora Martínez, Martín, . II. Universidad de la República, Programa de Posgrado en Ingeniería Matemática. III. Título.

[a]La investigación que da origen a los resultados presentados en la siguiente publicación recibió fondos de la Agencia Nacional de Investigación e Innovación bajo el código POS_NAC_2016_1_130953.

INTEGRANTES DEL TRIBUNAL DE DEFENSA DE TESIS

Prof. Dra. Paola Bermolen

Prof. Dr. Marcelo Fiori

Prof. Dr. Ignacio Ramírez

Montevideo – Uruguay

Abril de 2019

RESUMEN

En esta tesis presentamos algunos modelos para describir la altura y el ritmo de melodías musicales. Utilizamos dichos modelos para simular melodías con alturas y ritmo aleatorios. Presentamos además una estrategia de comparación de modelos basada en el principio del mínimo largo de descripción (MDL), que utilizamos para comparar los distintos modelos para ritmo estudiados.

Palabras claves:
Probabilidad, Estadística, Música, Cadenas de Markov, MDL, Simulación.

ABSTRACT

In this thesis we present some models to describe the pitch and rhythm of musical melodies. We use these models to simulate melodies with random pitches and rhythm. We also present a model comparison strategy based on the Minimum Description Length principle (MDL), which we use to compare the different rhythm models.

Keywords:
Probability, Statistics, Music, Markov Chains, MDL, Simulation.

Tabla de contenidos

Capítulo 1

Introducción

La relación entre matemática y música presenta diversos aspectos. A modo de ejemplo, el vínculo entre las frecuencias de las distintas alturas ha sido a lo largo de la historia un elemento clave a la hora de definir intervalos y escalas [BS03]. Asimismo el estudio de herramientas matemáticas para trabajar con fenómenos musicales presenta una gama amplia y variada de aplicaciones. Éstas se han incrementado notablemente con el desarrollo tecnológico que permite digitalizar el audio, así como procesar cantidades de datos cada vez mayores. Podemos considerar a grandes rasgos tres tipos de aplicaciones:

- Procesamiento de señales de audio. En este caso la música está dada en forma de señal digital de audio de la cual queremos extraer cierta información, por ejemplo determinar las frecuencias fundamentales de sus sonidos (que suelen caracterizar la altura de la nota percibida), identificar la estructura métrica subyacente y/o los patrones rítmicos presentes ([NRJB15], [Tem07]). Como objetivo más ambicioso se presenta incluso la transcripción completa de una melodía o pieza a notación simbólica (existe software específico destinado a este propósito, por ejemplo *AnthemScore* [Lun]).
- Síntesis de sonido. Recordemos que el sonido se compone de ondas las cuales se pueden generar matemáticamente. En el caso de los sonidos con alturas definidas tenemos ondas periódicas cuyas componentes sinusoidales se encuentran en relación armónica. Dichas componentes se presentan con diversa intensidad dando lugar al timbre característico de la fuente emisora. Así, el uso de funciones para generar sonidos cuyos timbres se asemejen a los de una fuente dada (por ejemplo un instrumento musical), o bien el diseño de nuevos timbres constituye un área de estudio en sí misma.

- Composición algorítmica. En este caso se utilizan diversos modelos matemáticos para producir música de forma automática, ya sea en forma de señal de audio o en notación simbólica. Este es el enfoque que motiva nuestro trabajo y que describimos con más detalle a continuación.

Existen diversas estrategias para automatizar total o parcialmente el proceso compositivo. Una de ellas consiste en incorporar elementos aleatorios en la composición. Por ejemplo, en la Música de Dados (Musikalisches Würfelspiel) del siglo XVII se propone el sorteo de fragmentos previamente compuestos de forma no automática, como se muestra en la figura 1.1. De este modo la incidencia del azar es limitada, en tanto los fragmentos a elegir fueron compuestos respetando, por ejemplo, una tonalidad y/o una armonía dadas. En el siglo XX compositores como John Cage y Iannis Xenakis incorporan también elementos aleatorios en la composición. Pero en estos casos, las composiciones no necesariamente respetan una tonalidad o estructura métrica dada. Cage hizo uso del azar para determinar distintos aspectos en sus composiciones con el supuesto objetivo de despersonalizar lo más posible los eventos musicales. Aprovecha además el entonces reciente desarrollo de la computación para obtener grandes cantidades de valores (pseudo) aleatorios para utilizarlos en la composición [Dic06]. Xenakis, por otra parte, utiliza diversos elementos matemáticos, varios de ellos aleatorios, en sus composiciones dando origen al término *música estocástica*.

Por otra parte, la posibilidad de generar y/o procesar grandes volúmenes de datos permitió desarrollar técnicas de generación automática de música que "aprenden" del repertorio existente e intentan emular su estilo, entendido como aquellos rasgos que caracterizan el *corpus* con el cual el sistema fue entrenado. Numerosos trabajos se han realizado en este sentido, entre los que destacamos el aporte de David Cope con Experiments in Music Intelligence [Cop96]. Su trabajo da lugar a la compositora ficticia Emily Howell [Cop], que genera música siguiendo ciertas formas (valses, fugas, preludios) y/o emulando a otros compositores. También destacamos el trabajo de François Pachet y el proyecto *Flow Machines* anteriormente dirigido por él. En este trabajo utilizamos un modelo basado en cadenas de Markov con restricciones también propuesto por Pachet ([PRB11], [Pac]). En la misma línea de simular música según estilos dados, pero utilizando en este caso modelos basados en redes neuronales se encuentra el trabajo de Daniel Jonhson([JKW17], [Joh]).

Nuestro trabajo está inicialmente motivado por la interrogante ¿Hasta qué punto es posible aprender los rasgos característicos de un estilo dado a partir de un conjunto de obras (*corpus*) representativo del mismo? Realizar exitosamente tal aprendizaje permitiría

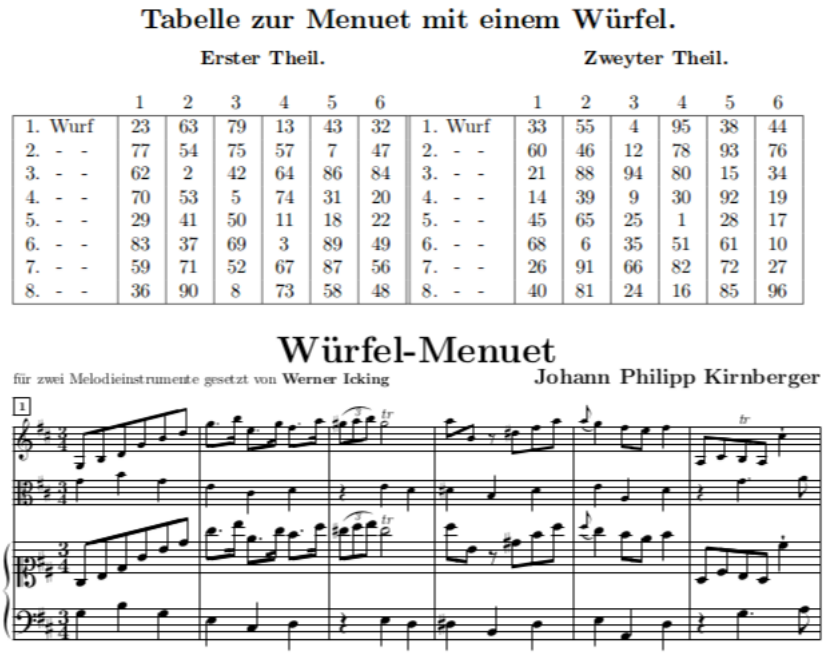

Tabelle zur Menuet mit einem Würfel.

	Erster Theil.							Zweyter Theil.					
	1	2	3	4	5	6		1	2	3	4	5	6
1. Wurf	23	63	79	13	43	32	1. Wurf	33	55	4	95	38	44
2. - -	77	54	75	57	7	47	2. - -	60	46	12	78	93	76
3. - -	62	2	42	64	86	84	3. - -	21	88	94	80	15	34
4. - -	70	53	5	74	31	20	4. - -	14	39	9	30	92	19
5. - -	29	41	50	11	18	22	5. - -	45	65	25	1	28	17
6. - -	83	37	69	3	89	49	6. - -	68	6	35	51	61	10
7. - -	59	71	52	67	87	56	7. - -	26	91	66	82	72	27
8. - -	36	90	8	73	58	48	8. - -	40	81	24	16	85	96

Würfel-Menuet

für zwei Melodieinstrumente gesetzt von **Werner Icking** **Johann Philipp Kirnberger**

Figura 1.1: Parte de la partitura de *Würfel-Menuet*, de J.P. Kirnberger y tabla con los números de compás a usar en cada paso según el resultado del dado. Los posibles compases a usar se encuentran preestablecidos de modo de preservar cierta estructura armónica. La obra completa puede encontrarse en [Wur].

luego crear de forma automática nuevas obras que respeten dicho estilo. Así, estudiamos modelos aleatorios para generar música que de algún modo se ajuste a ciertos corpus preexistentes. Retomamos y ampliamos lo estudiado en [Rum17], donde se propuso una estrategia para simular alturas que asignamos a una rítmica preestablecida para obtener melodías. A esto agregamos la discusión de distintos posibles modelos para simular también el ritmo.

Ante la diversidad de modelos posibles se desprende otra interrogante: ¿Cómo elegimos entre distintos modelos? Queremos disponer de un criterio de evaluación más allá de la apreciación subjetiva de los ejemplos simulados, pregunta que motiva la parte central de esta tesis.

Dedicamos parte de nuestro trabajo a presentar los modelos utilizados para simular las melodías y a detallar su implementación: para las alturas utilizamos cadenas de Markov con restricciones, mientras que para las duraciones exploramos algunos modelos propuestos por Temperley [Tem10] proponiendo además un par de modelos alternativos. Utilizamos algunos de los modelos descritos para generar melodías aleatorias, simulando de forma independiente las alturas y la duración.

La parte central de esta tesis consiste en la presentación y aplicación de dos posibles estrategias para comparar modelos, ambas basadas en el principio del mínimo largo de descripción (MDL, por su nombre en inglés). La primera, conocida como *Crude Two-Part MDL* nos permite comparar modelos de forma sencilla y con penalización por sobreajuste. La segunda es una versión refinada de Crude Two-Part MDL, cuyo fundamento es similar al planteado por Barron, Rissanen y Yu en [BRY], que contempla y procura enmendar algunas falencias del primer enfoque.

Tanto para la simulación de ejemplos como para la comparación de modelos estimamos los parámetros de cada modelo para el ritmo a partir de la rítmica de ciertos corpus preestablecidos. Nuestra elección se ve limitada en tanto la disponibilidad de obras en un formato adecuado es escasa. Siendo que cada corpus debe tener la misma estructura métrica para que la estimación tenga sentido, consideramos un total de cuatro conjuntos de obras en $\frac{4}{4}$ y tres en $\frac{2}{4}$. La mayoría de éstos son corpus de melodías folk europeas, de fácil acceso. Con el fin de darle una impronta local al trabajo incorporamos también un corpus de tangos canción.

Cabe destacar que en este trabajo consideramos, tanto para el análisis como para la simulación, notación musical simbólica. Es decir que no nos competen los tópicos de los dos primeros ítems descritos al principio de esta introducción, los cuales se enmarcan

dentro del procesamiento de audio, sino la información que podamos obtener a partir de música escrita de modo que podamos identificar fácilmente la altura y duración de sus notas. En este mismo formato se simulan nuestros ejemplos musicales.

Como contribuciones al área de esta tesis queremos destacar el enfoque propuesto para el cálculo del largo de descripción, en la versión refinada de Crude Two-Part MDL detallada en el capítulo 3.[1] Por otra parte resaltamos la presentación de un corpus nuevo, poco conocido debido a su reciente publicación, que contempla un género sobre el cual no se encontraban bases de datos (debidamente curadas) en notación simbólica.

La tesis se organiza de la siguiente manera: el capítulo 2 está dedicado a presentar una posible estrategia para simular melodías con altura y ritmo aleatorios, describiendo los modelos utilizados para ello así como otros modelos para ritmo. En el capítulo 3 presentamos criterios para comparar el ajuste de distintos modelos a un conjunto de datos dado. Estos se aplican a los distintos modelos estudiados y a distintos corpus musicales en el capítulo 4. Además incorporamos un anexo con más ejemplos y un repositorio online con los ejemplos en audio y el código utilizado [rep].

[1] Esta propuesta sigue una sugerencia de Ignacio Ramírez.

Capítulo 2

Simulación de melodías

2.1. Consideraciones preliminares

Dedicamos este capítulo a presentar algunas nociones de teoría musical que necesitamos para el resto del trabajo, así como los modelos que utilizamos en las posteriores simulaciones y análisis.

En una visión muy simplificada de la música, asumimos para nuestro estudio que un sonido está determinado por dos parámetros:

- **Altura:** Considerando el sonido como una onda periódica que se compone de distintas sinusoides en relación armónica, la altura queda determinada por la *frecuencia fundamental* de ésta. Auditivamente, es la cualidad que nos permite ordenar los sonidos de *grave* a *agudo*.
- **Duración:** es la extensión del sonido en el tiempo.

Suponemos que los valores que estos parámetros toman son discretos. Llamamos *nota* a un par (altura, duración) y *melodía* a una secuencia de notas. Presentamos a continuación algunos conceptos básicos útiles para el desarrollo del resto del trabajo.

2.1.1. Alturas, clases de altura y acordes

Como se mencionó antes, asumimos que el conjunto de alturas posibles es discreto, y se identifica cada una de ellas con un nombre. Si bien no nos interesa profundizar

sobre la nomenclatura, cabe observar que el nombre que una altura recibe no es único ni la identifica unívocamente ya que distintas alturas pueden tener el mismo nombre: Consideraremos que el conjunto de alturas posibles puede particionarse en *octavas*, cada una de las cuales tiene 12 alturas distintas (y con distinto nombre) consecutivas, como puede observarse en 2.1.[1] Así, si queremos distinguir dos alturas con igual nombre debemos también indicar la octava en que se encuentran. En la figura 2.2, por ejemplo, tenemos varios fa en distintas octavas.

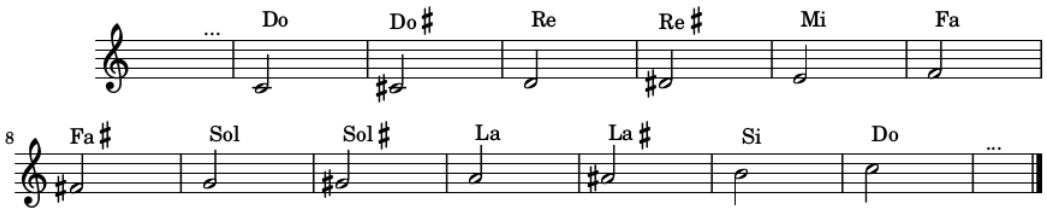

Figura 2.1: Notas de la escala cromática con sus correspondientes nombres, utilizando sostenidos en todos los casos.

Observación. Podemos definir una relación entre alturas, donde dos de ellas están relacionadas si y sólo si tienen igual nombre. Dicha relación es una relación de equivalencia cuyas clases de equivalencia se denominan *clases de altura*. La agrupación en octavas no es arbitraria: las alturas con igual clase de altura son altamente consonantes, por lo que para algunos propósitos resulta más adecuado considerar la clase de altura sin distinguir representantes.

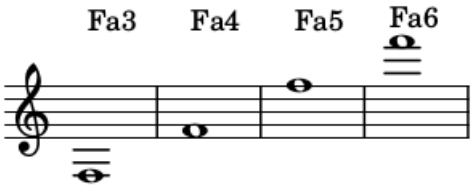

Figura 2.2: Cuatro notas *fa* en distinas octavas que se indican con la numeración correspondiete a la Scientific Pitch Notation ([Sci]). Si bien esta notación es convencional, en la práctica suele omitirse el número a no ser que sea necesario indicar explícitamente la octava.

Por otra parte, la superposición de (usualmente 3 o más) alturas da lugar a *acordes*, cuya organización constituye la *armonía* de la música. En este trabajo consideramos música monofónica y *tonal*, es decir que sus notas están fuertemente jerarquizadas distribuyendo

[1] Este es el denominado *sistema de afinación igualmente temperado de 12 alturas*, donde cada altura tiene más de un nombre posible. Omitiremos una discusión más profunda de las notas enarmónicas pues no resulta de particular interés en este trabajo.

sus roles en torno a una altura principal denominada *tónica*. Pese a que nuestras melodías no tienen notas simultáneas, hay una *armonía* subyacente. Es decir, es posible inferir de la melodía los distintos acordes que rigen en cada momento aunque no estén presentes de forma explícita.[1] Determinar tales combinaciones de acordes suele no ser trivial y no contamos con una estrategia para hacerlo de forma automática, por lo que en caso de utilizar la armonía, necesitamos tenerla dada o anotarla manualmente.

No es nuestro objetivo discutir en este trabajo estrategias para determinar los acordes ni las reglas para combinarlos en la música tonal. Simplemente será de utilidad notar que las alturas presentes en la melodía están relacionadas con la armonía siendo preferentes (en tanto se perciben como más consonantes) las alturas que pertenecen al acorde. Las demás alturas suelen utilizarse supeditadas a las primeras.

2.1.2. Duraciones, compases y fuerza métrica

Presentamos a continuación algunas nociones y supuestos que hacemos en nuestros modelos para ritmo, entendiendo por ritmo los patrones de duración presentes en la música. Para ello consideramos que éstos pueden dividirse en *compases*, que entendemos como fragmentos rítmicos de igual duración y con cierta estructura interna.

El término compás refiere tanto a la indicación de tales grupos como a los grupos en sí, y se anota con una fracción que indica la cantidad de unidades (numerador) de una cierta figura (denominador). Así, por ejemplo, la notación $\frac{2}{4}$ indica que cada compás abarca la duración equivalente a dos negras, mientras que $\frac{6}{8}$ refiere a compases con duración de 6 corcheas.

Lendarhl y Jackendoff [LJ83] proponen una jerarquía métrica que estructura los patrones rítmicos de acuerdo una grilla métrica cuyos elementos llamaremos *beats* o *pulsos*, entendidos como instantes equiespaciados en el tiempo. Así, si bien los pulsos carecen de duración, podemos identificarlos con la duración del lapso transcurrido ente un pulso y el siguiente. A modo ilustrativo, un compás de $\frac{4}{4}$ puede dividirse en 8 pulsos de corchea, o 4 pulsos de negra, ente otras opciones. Así, tenemos distintos *niveles* de subdivisión, de cuya superposición se desprende la idea de *fuerza métrica*, como se ilustra en la figura 2.3.

Sabemos que los patrones rítmicos en ciertos tipos de música se corresponden con una estructura métrica que les brinda un marco de referencia. En este trabajo consideraremos música compuesta con una estructura métrica definida y que conocemos (ya que dispon-

[1] Entendemos por acorde un conjunto de (generalmente tres o más) alturas.

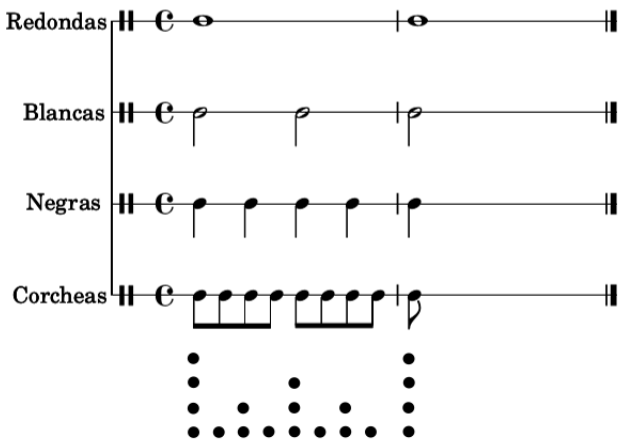

Figura 2.3: Subdivisión del compás en distintos niveles métricos. Los puntos representan las fuerzas métricas resultantes.

dremos de la música anotada en un formato simbólico), con lo cual no nos ocuparemos del problema de inferir la estructura, sino de explicar los patrones rítmicos observados en función de ésta.

Entendemos pues que un buen modelo probabilístico para ritmo debería asignar probabilidad alta a los patrones usuales de la estructura y estilo estudiados y probabilidad baja a los que son atípicos. Esto no implica necesariamente que el ritmo sea aleatorio (aunque los modelos puedan utilizarse para simular música), sino que utilizamos un enfoque probabilista para tratar de comprender mejor los rasgos más usuales del estilo. Veremos a continuación algunos aspectos teóricos y supuestos realizados para facilitar la construcción de los modelos.

En primer lugar consideraremos una unidad mínima de duración (por ejemplo, la corchea) y subdividiremos cada compás respecto a ella. Llamaremos *pulsos* a cada una de dichas unidades e identificaremos cada secuencia rítmica con los pulsos en los que hubo un ataque (ocurrencia de una nota). De este modo cada secuencia puede representarse como una secuencia de unos y ceros indicando los lugares de ataque y no ataque respectivamente. Cabe tener en cuenta que esta representación implica algunas omisiones:

- Lo único que podemos distinguir es si en un pulso dado ocurrió o no un ataque. En particular, representamos de igual modo una nota larga y una corta sucedida por silencios, como puede observarse en la figura 2.4. Tampoco podemos hacer distinciones sobre los distintos tipos de acento, dado por aspectos como la dinámica en la interpretación o el rol que la nota cumple en la tonalidad subyacente (en caso

de que haya una melodía). Estos elementos también aportan información sobre la estructura métrica que nuestros modelos no tendrán en cuenta.

- No podemos representar ataques que no coincidan con algún pulso (lo cual limita el uso de duraciones menores a un pulso). La duración elegida para el pulso deberá ser entonces lo suficientemente pequeña para representar las secuencias rítmicas que se desee analizar.

A modo de ejemplo, bajo esta representación un compás de $\frac{4}{4}$ subdividido en pulsos de corchea consiste en una secuencia de ocho números (unos y ceros). Un compás de $\frac{3}{4}$ subdividido en semicorcheas, en cambio, se traduce en una secuencia de doce números.

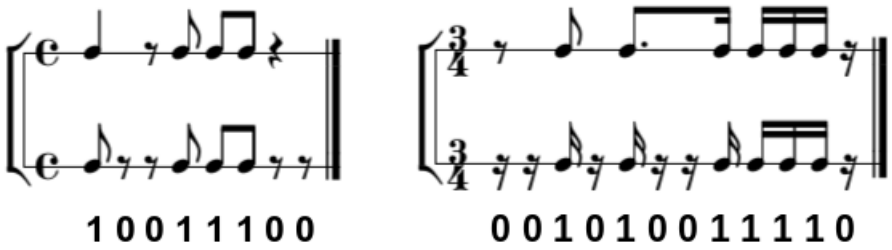

Figura 2.4: Un compás de $\frac{4}{4}$ y otro de $\frac{3}{4}$, subdivididos en corcheas y semicorcheas respectivamente. En ambos casos se indican los lugares de ataque y las secuencias de unos y ceros asociadas.

Tenemos así una forma de representar secuencias rítmicas como listas de números, los cuales podemos pensar como valores tomados por sucesiones de variables aleatorias cuyo comportamiento (distribución y dependencia entre ellas) queremos describir. Sin entrar aún en detalles podemos observar que no esperamos que sean independientes e idénticamente distribuídas (i.i.d.), en tanto se observa que las variables correspondientes al primer pulso del compás, por ejemplo, tienen mayor probabilidad de ser 1 (es decir, de presentar un ataque) que las que corresponden a pulsos más débiles. En consecuencia puede resultar de utilidad considerar distintos niveles de *fuerza métrica* en la construcción de los modelos.

En este sentido consideraremos lo propuesto por Barlow en [Bar87], donde se establece una relación jerárquica entre los distintos pulsos de un compás (a los que también referiremos como posición métrica, en tanto está supeditada al compás). Queda así definida una *grilla métrica* con distintos niveles de fuerza métrica. Temperley se basa en esta noción

para definir en [Tem07] grillas métricas para algunos compases usuales, como podemos observar en la figura 2.5. Nótese que los niveles son coherentes con una noción intuitiva de pulso fuerte o débil intrínseca al compás.

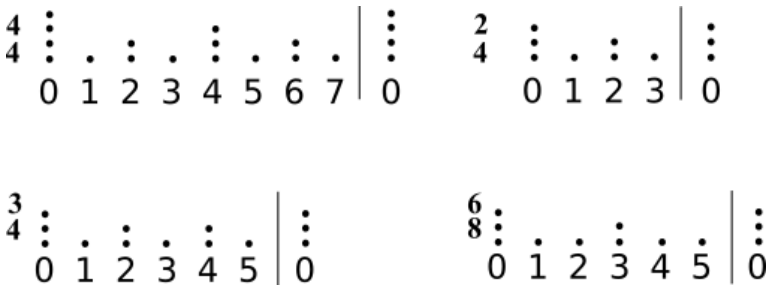

Figura 2.5: Niveles métricos para compases de $\frac{4}{4}$, $\frac{3}{4}$, $\frac{2}{4}$ y $\frac{6}{8}$, todos subdivididos en pulsos de corchea. Los números indican la posición dentro del compás y los puntos el nivel métrico de cada posición. La cantidad de puntos indica el nivel, donde niveles más grandes corresponden a pulsos más fuertes.

Bajo los supuestos aquí mencionados presentamos y comparamos en la siguiente sección algunos posibles modelos para la estructura métrica. Comenzaremos por los 6 modelos propuestos por Temperley en [Tem10] para luego proponer algunas alternativas.

2.2. Modelos para ritmo

2.2.1. Los modelos de Temperley

De acuerdo a lo establecido anteriormente, representaremos las secuencias rítmicas como secuencias de unos y ceros que a su vez son observaciones de variables aleatorias. Así, dada una secuencia de N pulsos $(x_0, x_1, \ldots, x_{N-1}) \in \{0,1\}^N$, asumiremos que fue generada por un vector de variables aleatorias $(X_0, X_1, \ldots, X_{N-1})$ cuya distribución de probabilidad depende del modelo.

Consideremos ahora un conjunto D de secuencias rítmicas a analizar, por ejemplo una obra musical o un conjunto de obras. Si dichas secuencias corresponden a un conjunto razonablemente homogéneo, es de esperar que algunos patrones rítmicos sean mucho más comunes que otros. Así, se construyen modelos apuntando a maximizar la probabilidad de ocurrencia de las secuencias observadas. Con ese objetivo Temperley propone en [Tem10] los siguientes 6 modelos paramétricos:

$\mathbf{M_1}$ *Modelo uniforme en posición.* Se asume que las secuencias son generadas por varia-

bles Bernoulli i.i.d. cuya probabilidad $p = \mathbf{P}(X_k = 1)$ representa la probabilidad de que haya un ataque en un pulso cualquiera. Si bien no parece ser un modelo fidedigno para la estructura métrica de la mayor parte de la música (desconoce por completo la existencia de compases, y en particular su estructura interna), es un modelo muy simple que sólo requiere un parámetro.

$\mathbf{M_2}$ *Modelo de duraciones independientes (o de orden cero).* En este caso consideramos que el tiempo (en pulsos) transcurrido entre ataques es una secuencia de variables i.i.d..

Para este modelo y algunos otros resulta conveniente considerar la siguiente representación alternativa: dada la secuencia $(x_0, \ldots, x_{N-1}) \in \{0,1\}^N$ definimos la secuencia auxiliar $(y_1, \ldots y_M)$ de modo que

$$y_j = \text{ Cantidad de ceros entre el } j\text{-ésimo 1 y el } j+1\text{-ésimo 1}, \tag{2.1}$$

asumiendo que las secuencias comienzan en 1.

Así, $M+1$ es la cantidad de unos en la secuencia $(x_i)_{i\in\{0,\ldots,N-1\}}$ y las variables Y_j representan los *tiempos interataque*.[1]

Por ejemplo, la secuencia de 8 pulsos $(1,0,1,0,0,0,1,1)$ da lugar a la secuencia de tiempos interataque $(1,3,0)$. Recíprocamente, si asumimos que la secuencia comienza con un 1 podemos recuperar el vector original conociendo su largo y el vector auxiliar (y_j).

Definiremos pues el segundo modelo a través de la distribución de $(Y_1, \ldots, Y_M)$. Dado que se asume que las variables Y_j son independientes, nos basta con conocer el vector (finito-dimensional) de probabilidades θ con $\theta_i := \mathbf{P}(Y_j = i)$.

Nótese que este modelo parece también bastante pobre, ya que al igual que el anterior no utiliza información de la estructura del compás.

$\mathbf{M_3}$ *Modelo de posición métrica.* En cada pulso se decide si hay o no un ataque con probabilidad que depende de los niveles métricos a los que pertenece el pulso en cuestión. A modo de ejemplo, si tenemos una secuencia rítmica en un contexto de $\frac{4}{4}$ subdividido en pulsos de corchea, habrá en total 4 probabilidades de ataque correspondientes a cada uno de las 4 fuerzas métricas que se tienen en el compás (ver figura 2.5), a saber:

- El primer pulso (posición métrica 0) de cada compás es el único presente en el nivel de redondas (fuerza 4). Allí ocurrirá un ataque con probabilidad θ_{IV}. En términos de los vectores aleatorios (X_j) definidos como antes, esto implica

[1] Usualmente denominados *Inter-Onset Inetrvals* (IOI) en inglés.

$$\mathbf{P}(X_j = 1) = \theta_{IV},\ \forall j = 0 \pmod 8.$$

- Hay también un único pulso en el nivel de blancas (mas no de redondas), en la posición métrica 4 del compás. Si llamamos θ_{III} a la probabilidad de ocurrencia de un ataque en él, se tiene

$$\mathbf{P}(X_j = 1) = \theta_{III},\ \forall j = 4 \pmod 8.$$

- Los pulsos de fuerza 2 se encuentran en las posiciones 2 y 6 y los de fuerza 1 en las posiciones $1, 3, 5$ y 7. Si θ_{II} y θ_I son sus respectivas probabilidades de ataque tenemos

$$\mathbf{P}(X_j = 1) = \theta_{II},\ \forall j \in \{2, 6\} \pmod 8,$$
$$\mathbf{P}(X_j = 1) = \theta_I,\ \forall j \in \{1, 3, 5, 7\} \pmod 8.$$

Notemos que las ocurrencias de ataque son independientes entre sí al igual que en los dos modelos anteriores, pero en este caso sí se utiliza la información proporcionada por la estructura métrica del compás. En general, la cantidad de parámetros será la cantidad de niveles considerados, que dependen de la subdivisión elegida.

$\mathbf{M_4}$ *Modelo de posición métrica refinado.* En cada pulso se decide si hay o no un ataque con probabilidad que depende de la posición de dicho pulso en el compás. Se trata de un modelo similar al anterior pero refinado en tanto no agrupamos los pulsos según los niveles a los que pertenece sino según su posición en el compás. Si nuevamente consideramos el contexto de $\frac{4}{4}$ con pulsos de corchea, tendríamos 8 probabilidades posiblemente distintas, una para cada posición del compás. Es decir

$$\mathbf{P}(X_j = 1) = \theta_i, \quad \text{siendo } i = j \pmod 8,$$

con $i \in \{0, \ldots, 7\}$.

En particular puede observarse que, para un mismo conjunto de datos, tenemos que bajo este modelo, θ_0 coincide con la probabilidad de nivel 4 en el modelo anterior, y θ_4 con la de nivel 3.

$\mathbf{M_5}$ *Modelo Jerárquico.* En cada pulso se tiene un ataque con probabilidad que depende de su fuerza métrica y está condicionada a si hay o no ataques en primeros pulsos de nivel superior a cada lado. Para ello definiremos el *tipo de anclaje* de un pulso, a saber:

- Decimos que un pulso es *no-anclado* si en ninguno de los dos pulsos de nivel

superior a cada lado hay un ataque.

- Un pulso es *pre-anclado* (*post-anclado*) si de ambos primeros pulsos de nivel superior sólo el anterior (posterior) tiene un ataque.
- Llamaremos *bi-anclados* a los pulsos con ataques en ambos pulsos de nivel superior a cada lado.

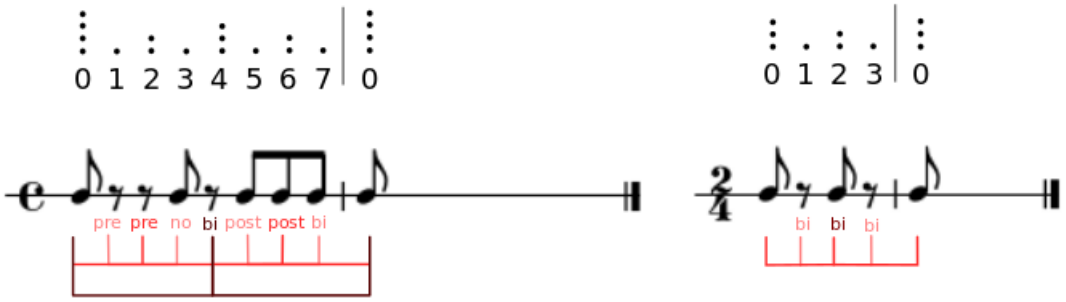

Figura 2.6: Un compás de $\frac{4}{4}$ y otro de $\frac{2}{4}$, indicando el tipo de anclaje de cada pulso y los vecinos de nivel superior más cercanos de los cuales éste se desprende.

En la figura 2.6 se pueden observar dos compases de ejemplo, indicando los tipos de anclaje de todos sus pulsos. Para facilitar la comprensión describamos la generación bajo este modelo de un compás de $\frac{4}{4}$ subdividido en corcheas:

- En primer lugar se establecen los ataques en los pulsos de nivel mayor (en este caso, de nivel 4, correspondientes al primer pulso de cada compás) asumiendo que son i.i.d.. Concretamente, si tenemos el conjunto de variables $\{X_j\}_{j\in\{0,\dots N\}}$, el subconjunto $\{X_j\}_{j\equiv 0 \ (mod\ 8)}$ está formado por variables i.i.d con distribución $Ber(\theta_{IV})$.
- A continuación se determinan los ataques en los pulsos de nivel inmediatamente inferior. La probabilidad de que ocurra un ataque en ellos depende del tipo de anclaje, por lo que hay que observar si hubo o no ataques en los pulsos de nivel superior adyacentes. En nuestro ejemplo son los pulsos de nivel 3 y corresponden a la posición 4 de cada compás. Su tipo de anclaje depende de lo que ocurra en los pulsos de nivel 4 adyacentes, es decir, la posición 0 del mismo compás y del siguiente. Nótese que las variables correspondientes a dichos pulsos ($\{X_j\}_{j\equiv 4 \ mod\ 8}$) no son independientes.

- Se procede de modo similar con los niveles restantes. Los pulsos de nivel 2 son, en este caso, los correspondientes a las posiciones 2 y 6 del compás. Sus tipos de anclaje quedan determinados por los pulsos de posición 0 y 4, y 4 y 0 (del compás siguiente) respectivamente. Finalmente se determinan los ataques de nivel 1, correspondientes a los pulsos de posición 1, 3, 5 y 7.

En la figura 2.7 se representa esta jerarquía entre las distintas posiciones del compás. Nótese que bajo este modelo, en nuestro ejemplo de $\frac{4}{4}$ con pulsos de corchea se tienen 13 parámetros a estimar.

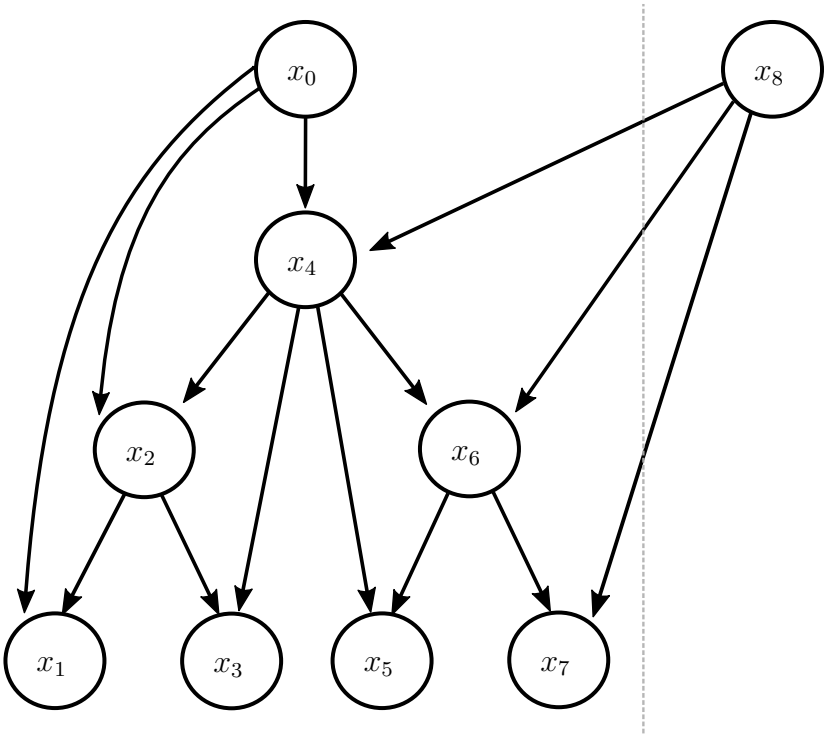

Figura 2.7: Grafo de dependencias entre las distintas posiciones métricas. Las flechas indican qué posiciones son las que determinan el tipo de anclaje.

Podemos ver en la figura 2.8 el esquema de generación aplicado a los compases de ejemplo de la figura 2.6. Nótese que los ataques o silencios que se agregan en cada paso tienen tipo de anclaje determinado por los que se agregaron en el paso anterior.

$\mathbf{M_6}$ *Modelo de posición de orden uno.* Se determina la posición métrica de cada ataque, condicionada a la del ataque anterior. Observemos que dado $j > 1$ la posición métrica del j-ésimo ataque puede escribirse como

$$S_j = \sum_{i=1}^{i=j-1} (Y_i + 1) \pmod 8,$$

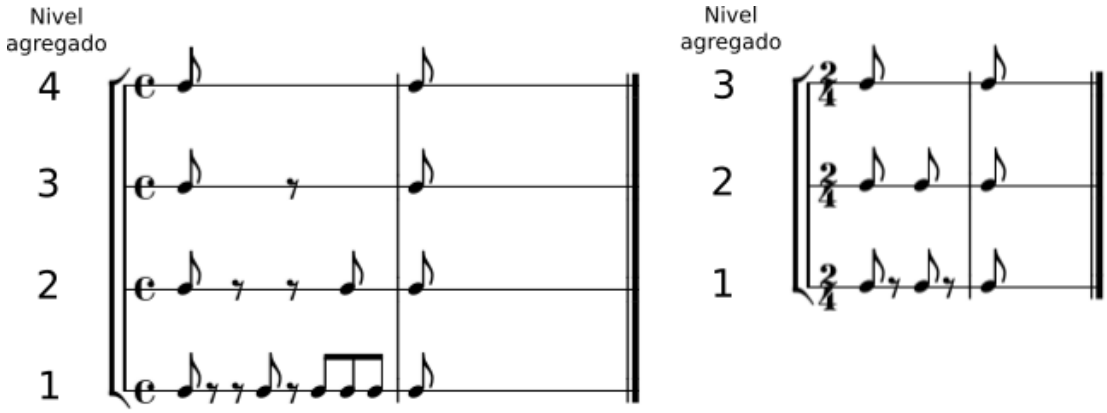

Figura 2.8: Generación paso a paso de las secuencias de la figura 2.6.

donde Y_i se define como en el modelo 2. Luego las variables S_j son generadas por una cadena de Markov (de orden 1) con espacio de estados finito (en el ejemplo que venimos considerando sería $\{0, \ldots 7\}$). Suele resultar un modelo con muchos parámetros (toda la matriz de transición), pese a que en la práctica es de esperar que muchos de ellos sean cero.

Modelo / Compas	$\frac{2}{4}$	$\frac{3}{4}$	$\frac{4}{4}$	$\frac{6}{8}$
1- Uniforme en posición	1	1	1	1
2- Duraciones independientes (considerando duraciones no mayores a un compas)	3	5	7	5
3- De posición métrica	3	3	4	3
4- De posición métrica refinado	4	6	8	6
5- Jerárquico	9	9	13	9
6- De posición de orden 1. (considerando duraciones no mayores a un compás)	12	30	56	30

Tabla 2.1: Cantidad de parámetros de cada uno de los modelos propuestos por Temperley para distintos compases. Se consideran en todos los casos pulsos de corchea.

2.2.2. Modelo jerárquico refinado

Hemos considerado hasta ahora 6 modelos, que podemos separar en dos grupos según si utilizan o no información relativa a la estructura del compás subyacente. Los modelos M_3,

M_4, M_5 y M_6 usan esta información de distintas formas: M_4 y M_6 definen sus parámetros basados en la posición métrica de cada pulso, mientras que M_3 y M_5 utilizan además los niveles métricos presentados en la sección 2.1.2. Por otra parte los modelos M_1 y M_2 no aprovechan la existencia de compases siendo, en este sentido, los modelos mas pobres (luego veremos que esto se refleja en su desempeño al utilizarlos sobre distintos corpus).

La consideración de los niveles métricos nos ofrece un criterio para agrupar los pulsos en distintos niveles, obteniendo así modelos con menos parámetros. Compárense por ejemplo los modelos M_3 y M_4, que tienen un fundamento similar pero difieren en el grado de refinamiento. El agrupamiento realizado en M_3 puede ser útil cuando el costo relativo de cada parámetro es muy alto (por ejemplo por tener un conjunto no tan grande de datos), pero poco conveniente en el caso contrario. De modo similar, se puede proponer construir una versión "refinada" del modelo M_5, es decir, en lugar de considerar los pulsos agrupados según su nivel (para luego condicionar a 4 posibles tipos de anclaje distintos) considerar otro tipo de agrupamiento.

Este enfoque es coherente con lo propuesto por Barlow en [Bar87], donde se presenta el concepto de *relevancia métrica* o *indispensabilidad*, que nos permitirá ordenar los pulsos incluso dentro de un mismo nivel. Intuitivamente podemos pensar la indispensabilidad como el ordenamiento de los pulsos que resulta de, a partir de un compás vacío, ir agregando ataques uno por uno de modo que el ritmo resultante refuerce lo más posible la estructura métrica subyacente. Para ello se asigna a cada pulso un número distinto entre 0 (menos relevante) y $n_{bpc}-1$ (más relevante), siendo n_{bpc} la cantidad de pulsos por compás. La formula general para el cálculo de indispensabilidades en cualquier estructura métrica puede encontrarse en [Bar87], aquí nos limitaremos a describirlo para algunos compases usuales.

- Primero se asignan valores de indispensabilidad a los pulsos de nivel más alto, siendo siempre el primer pulso del compás el de mayor indispensabilidad.
- Para los sucesivos niveles inferiores los pulsos se ordenan de acuerdo a la indispensabilidad del primer pulso de nivel mayor que les suceda, asignando indispensabilidad mayor a aquellos que preceden a pulsos más relevantes.

De este modo todos los pulsos del compás quedan jerarquizados preservando el orden de los niveles métricos. En los parámetros calculados en los experimentos presentados en el capitulo 4 se observa que la indispensabilidad así definida es coherente con las proporciones de ataque en cada pulso observadas en los distintos corpus estudiados. Por otra parte notemos que, por ejemplo, en los niveles más bajos (que agrupan mayor cantidad

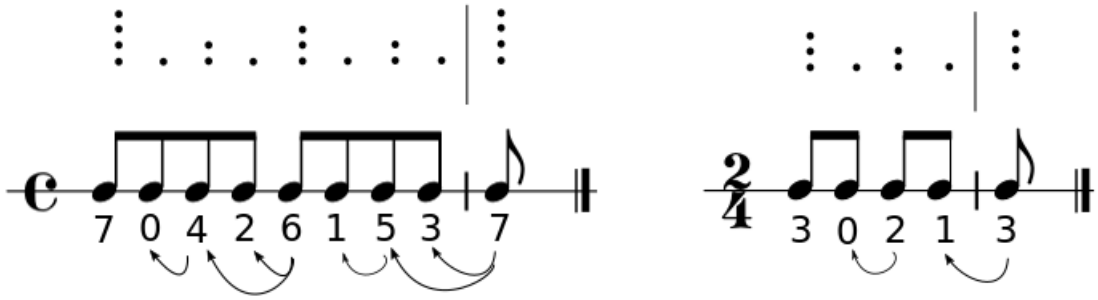

Figura 2.9: Indispensabilidades para compases de $\frac{4}{4}$ y $\frac{2}{4}$. Las flechas indican el pulso posterior de referencia para los pulsos que no son únicos en su nivel.

de pulsos) la indispensabilidad llega a diferir bastante entre un pulso y otro, lo cual puede afectar también a las probabilidades según los distintos tipos de anclaje.

Por ejemplo podemos ver en la figura 2.9 que para el compás de $\frac{4}{4}$ con pulsos de corchea los pulsos de nivel 1 tienen indispensabilidades desde 0 hasta 3. Sin embargo, el modelo M_5 les asigna a todos ellos las mismas probabilidades condicionales. Así, los patrones presentados en la figura 2.10 tienen la misma probabilidad bajo el modelo M_5, mientras que observaremos en nuestros corpus que las tasas de ataque en el segundo y penúltimo pulso del compás difieren significativamente.

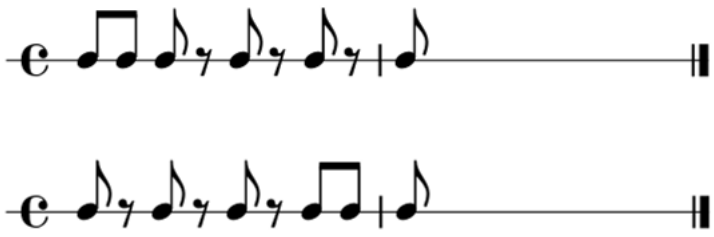

Figura 2.10: Dos patrones con igual probabilidad bajo M_5.

Por otra parte utilizar el modelo M_4, que sí distingue pulso a pulso, nos hace perder la información relativa a los niveles métricos y los tipos de anclaje. Esto no parece conveniente en tanto estas nociones están relacionadas con la relevancia métrica propuesta por Barlow. En base a estas consideraciones presentamos el siguiente modelo alternativo para ritmos. En primer lugar recordemos escuetamente cómo se definieron los parámetros en el modelo M_5.

- Se agrupan los distintos pulsos del compás según su nivel métrico. Se tienen así tantos grupos como niveles.
- Para cada uno de estos niveles, se consideran 4 probabilidades correspondientes a los distintos tipos de anclaje. Como excepción se tiene el pulso correspondiente al nivel más alto, para el cual se considera una única probabilidad de ataque.

Los modelos que propondremos modifican la forma en que se agrupan los pulsos. Consideraremos los siguientes dos agrupamientos:

1. Agrupamos los pulsos según su nivel métrico como en M_5, con excepción de los correspondientes al nivel 1 (nivel con mayor cantidad de pulsos). Éstos se consideran individualmente. Los grupos resultantes son:
   - Un grupo para cada nivel métrico, desde el mayor nivel hasta el nivel 2.
   - Un grupo más para cada pulso de nivel 1.
2. Consideramos cada pulso individualmente, es decir, se tienen tantos grupos como pulsos.

Observación. En algunos compases (por ejemplo $\frac{2}{4}$) ambas propuestas son equivalentes. Esto ocurre cuando el único nivel con más de un pulso es el nivel 1.

Es claro que algunos de los modelos propuestos hasta ahora parecen más adecuados que otros, en tanto aprovechan mejor la teoría musical subyacente. En el caso de M_1 y M_2, por ejemplo, difícilmente obtengamos de ellos un buen insumo para analizar ningún corpus musical. Sin embargo, más allá de esta intuición, no hemos definido aún ningún criterio para decidir qué entendemos por "buen modelo".

Ante un conjunto D de datos que queremos analizar y una familia de modelos paramétricos que asignan respectivas probabilidades a D, un primer criterio para determinar cuál de esos modelos es el que mejor se ajusta a D es considerar el modelo que asigne a D mayor probabilidad. Si bien veremos más adelante los inconvenientes de este enfoque, nos será de utilidad calcular las probabilidades correspondientes para cada uno de los modelos presentados.

2.2.3. Cálculo de las probabilidades

Sea D una secuencia $(x_0, \ldots, x_{N-1})$ de unos y ceros y $(y_1, \ldots, y_M)$ su correspondiente secuencia de tiempos interataque definidos como en la ecuación (2.1). Para facilitar la

comprensión, asumiremos que D proviene de un compás de $\frac{4}{4}$ subdividido de corcheas, siendo análogos los cálculos para otros compases o subdivisiones. Se tienen así 8 pulsos por compás y asumiremos $N = \dot{8}$, es decir, una cantidad entera de compases.

$\mathbf{M_1}$ Llamando $p := P(X = 1)$ a la probabilidad de ataque, se tiene

$$\mathbf{P}_{\mathbf{M_1}}(D \mid p) = p^{\sum_{i=0}^{N-1} x_i}(1-p)^{N-\sum_{i=0}^{N-1} x_i}.$$

Recordemos que el estimador máximo verosímil de p es $\hat{p} = \frac{1}{N}\sum_{i=0}^{N} x_i$, el cual se sustituye luego en la fórmula anterior en caso de querer maximizar la probabilidad.

$\mathbf{M_2}$ Sea $T := \text{máx}\{y_j\}$ el mayor tiempo interataque observado. Los parámetros a estimar son $\theta = \{\theta_j\}_{j=0,\dots,T}$ con $\theta_j = \mathbf{P}(Y = j)$, siendo Y la variable aleatoria que modela los tiempos interataque. Sus estimadores son

$$\hat{\theta}_j = \frac{1}{M+1}\sum_{i=1}^{M} \mathbb{1}_{\{y_i = j\}}. \tag{2.2}$$

Nótese que $\sum_{j=0}^{M} \theta_j = 1$. La probabilidad de D bajo θ conocido es

$$\mathbf{P}_{M_2}(D \mid \theta) = \prod_{j=0}^{T} \theta_j^{\sum \mathbb{1}_{\{y_i = j\}}}.$$

$\mathbf{M_3}$ Se tienen tantas probabilidades para estimar como niveles métricos se hayan definido (en nuestro caso, 4). Estas son

$$\theta_4 = \mathbf{P}(X_i = 1),\ \text{para } i \equiv 0 \pmod 8, \quad \theta_3 = \mathbf{P}(X_i = 1),\ \text{para } i \equiv 4 \pmod 8,$$
$$\theta_2 = \mathbf{P}(X_i = 1),\ \text{para } i \equiv 2 \pmod 4, \quad \theta_1 = \mathbf{P}(X_i = 1),\ \text{para } i \equiv 1 \pmod 2.$$

Sus correspondientes estimadores resultan

$$\hat{\theta}_4 = \tfrac{8}{N}\sum_{i=\dot{8}} x_i, \qquad \hat{\theta}_3 = \tfrac{8}{N}\sum_{i \equiv 4 \pmod 8} x_i,$$
$$\hat{\theta}_2 = \tfrac{4}{N}\sum_{i \equiv 2 (\text{mod } 4)} x_i, \quad \hat{\theta}_1 = \tfrac{2}{N}\sum_{i \equiv 1 \pmod 2} x_i.$$

Y los datos tienen probabilidad

$$\mathbf{P}_{M_3}(D \mid \theta) = \left(\prod_{i=\dot{8}} \theta_4^{x_i}(1-\theta_4)^{1-x_i}\right)\left(\prod_{\substack{i\equiv 4 \\ (\text{mod } 8)}} \theta_3^{x_i}(1-\theta_3)^{1-x_i}\right)$$
$$\left(\prod_{\substack{i\equiv 2 \\ (\text{mod } 4)}} \theta_2^{x_i}(1-\theta_2)^{1-x_i}\right)\cdot\left(\prod_{\substack{i\equiv 1 \\ (\text{mod } 2)}} \theta_1^{x_i}(1-\theta_1)^{1-x_i}\right).$$

$\mathbf{M_4}$ En este caso hay tantos parámetros como pulsos tenga cada compás (8 en el caso considerado). Éstos se definen como $\{\theta_j\}_{0\leq j<8}$ con

$$\theta_j = \mathbf{P}(X_i = 1), \text{ para } i \equiv j \pmod 8.$$

Sus estimadores son

$$\hat{\theta}_j = \frac{8}{N}\sum_{i\equiv j \pmod 8} x_i,\ j \in \{0,\dots 7\}$$

y la probabilidad de D resulta

$$\mathbf{P}_{M_4}(D \mid \theta) = \prod_{j=0}^{j=7}\left(\prod_{i\equiv j \pmod 8} \theta_j^{x_i}(1-\theta_j)^{1-x_i}\right).$$

$\mathbf{M_5}$ La cantidad de parámetros en este modelo es

$$4(\text{cantidad de niveles } -1)+1.$$

Esto nos da los siguientes 13 parámeros en nuestro caso de estudio:

$$\theta_{IV} = \mathbf{P}(X_i = 1), \text{ para } i \equiv 0 \pmod 8, \text{ (igual a } \theta_0 \text{ definida en el modelo 4)}$$
$$\theta_{k_{ij}} = \mathbf{P}(X_l = 1), i, j \in \{0,1\}, k \in \{I, II, III\},\ l \text{ pulso de nivel k,}$$

donde $00, 01, 10$ y 11 representan los tipos de anclaje no, post, pre y bi-anclado respectivamente.

Necesitamos definir algo más de notación. Definiremos

$n_{IV} = \sum_{i=\dot{8}} x_i$, cantidad de ataques en pulsos de nivel 4,

$n_{k_{ij}}$ = cantidad de ataques en pulsos de nivel k con tipo de anclaje ij, $i, j \in \{0, 1\}, k \in \{I, II, III\}$,

$n^0_{k_{ij}}$ = cantidad de no ataques en pulsos de nivel k con tipo de anclaje ij, $i, j \in \{0, 1\}, k \in \{I, II, III\}$.

Luego obtenemos los parámetros estimados por máxima verosimilitud y la probabilidad de D:

$$\begin{aligned}\hat{\theta}_{IV} =& \frac{8}{N} \sum_{i=\dot{8}} x_i \text{ (como antes)},\\ \hat{\theta}_{k_{ij}} =& \frac{n_{k_{ij}}}{n_{k_{ij}} + n^0_{k_{ij}}}, i, j \in \{0, 1\}, k \in \{I, II, III\}.\end{aligned}$$

$$\mathbf{P}_{M_5}(D \mid \theta) = \theta_{IV}^{n_{IV}} \left(1 - \theta_{IV}^{\frac{N}{8} - n_{IV}}\right) \prod_{k \in \{I,II,III\}} \prod_{i,j \in \{0,1\}} \theta_{k_{ij}}^{n_{k_{ij}}} \left(1 - \theta_{k_{ij}}\right)^{n^0_{k_{ij}}}.$$

$\mathbf{M_6}$ Los parámetros a estimar son las entradas de una matriz de transición cuya dimensión es 8×8 (o más en general $n_{bpc} \times n_{bpc}$ con n_{bpc} la cantidad de pulsos por compás). Así, tenemos $P = (p_{ij})_{i,j \in \{0,\dots,7\}}$ con

$$p_{ij} = \mathbf{P}(S_2 \equiv j_{(\text{mod } 8)} \mid S_1 \equiv i_{(\text{mod } 8)}),$$

y sus correspondientes estimadores

$$\hat{p}_{ij} = \frac{\sum_{l=1}^{l=M-1} \mathbb{1}_{\{S_{l+1}=j, S_l=i\}}}{\sum_{l=1}^{l=M-1} \mathbb{1}_{\{S_l=i\}}}.$$

La probabilidad de los datos observados resulta

$$\mathbf{P}_{M_6}(D \mid P) = \prod_{i=0}^{i=7} \prod_{j=0}^{j=7} p_{ij}^{n_{ij}},$$

siendo n_{ij} la cantidad de transiciones de i a j observadas en la secuencia de posiciones $\{s_l\}$, con $s_l = \sum_{h<l}(y_h + 1)\ mod n_{bpc}$.

Al igual que en los modelos de Temperley presentaremos las probabilidades para el compás de $\frac{4}{4}$ subdividido en corcheas, siendo similares los cálculos para otros compases y subdivisiones. Llamaremos M_7 al modelo menos refinado de los dos propuestos, y M_{total} al que considera todos los pulsos de forma individual. Tenemos pues:

$\mathbf{M_7}$ En $\frac{4}{4}$ se tiene un total de 25 parámetros, correspondientes a las probabilidades bi, pre, post y no-ancladas de los 6 grupos distintos (4 para los pulsos de nivel 1, 1 para el nivel 2, 1 para el nivel 3) más la probabilidad de ataque del nivel 4. En este caso denominaremos:

- $\theta_{k_{ij}}$ con $k \in \{1, 3, 5, 7\}$ y $i, j \in \{0, 1\}$ las probabilidades de ataque en el pulso de posición k-ésima en el compás, con tipo de anclaje ij (definido como en M_5). Estas son las probabilidades correspondientes a los pulsos de nivel 1.
- $\theta_{II_{ij}}$ y $\theta_{III_{ij}}$ las probabilidades con tipo de anclaje ij de nivel 2 y 3 respectivamente.
- θ_{IV} la probabilidad de ataque de nivel 4.
- $n_{IV} = \sum_{i=\dot{8}} x_i$ cantidad de ataques en pulsos de nivel 4.
- $n_{k_{ij}}$ es la cantidad de ataques en pulsos del grupo k-ésimo con tipo de anclaje ij, siendo $i, j \in \{0, 1\}, k \in \{1, 3, 5, 7, II, III\}$.
- $n^0_{k_{ij}}$, cantidad de no ataques en pulsos del grupo k-ésimo con tipo de anclaje ij, siendo $i, j \in \{0, 1\}, k \in \{1, 3, 5, 7, II, III\}$.[1]

Con esta notación, tenemos que el modelo M_7 asigna a nuestro conjunto de datos D la probabilidad

$$\mathbf{P}_{M_7}(D \mid \theta) = \theta_{IV}^{n_{IV}} \left(1 - \theta_{IV}^{\frac{N}{8} - n_{IV}}\right) \prod_{k \in \{1,3,5,7,I,II,III\}} \prod_{i,j \in \{0,1\}} \theta_{k_{ij}}^{n_{k_{ij}}} \left(1 - \theta_{k_{ij}}\right)^{n^0_{k_{ij}}}.$$

Nota: Recordar que $N = \#D$.

$\mathbf{M_{total}}$ Se tienen en este caso 29 parámetros. De modo similar al modelo anterior notaremos:

[1] Nótese que en este caso los parámetros homónimos en M_5 son efectivamente los mismos.

- $\theta_{k_{ij}}$ con $k \in \{1, \ldots, 7\}$ y $i, j \in \{0, 1\}$ las probabilidades de ataque en el pulso de posición k-ésima en el compás, con tipo de anclaje ij (definido como en M_5).
- θ_0, probabilidad de ataque en el pulso de posición 0 (es igual al θ_{IV} definido antes, lo renombramos por comodidad).
- $n_{k_{ij}}$, cantidad de ataques en pulsos de posición k con tipo de anclaje ij, siendo $i, j \in \{0, 1\}, k \in \{1, \ldots, 7\}$.
- n_0, cantidad de ataques en pulsos de posición 0.
- $n^0_{k_{ij}}$, cantidad de no ataques en pulsos de posición k con tipo de anclaje ij, siendo $i, j \in \{0, 1\}, k \in \{1, \ldots, 7\}$.

La probabilidad de D bajo este modelo resulta

$$\mathbf{P}_{M_{total}}(D \mid \theta) = \theta_0^{n_0} \left(1 - \theta_0^{\frac{N}{8} - n_0}\right) \prod_{k \in \{1,\ldots,7\}} \prod_{i,j \in \{0,1\}} \theta_{k_{ij}}^{n_{k_{ij}}} \left(1 - \theta_{k_{ij}}\right)^{n^0_{k_{ij}}}.$$

En la seccion 2.4.1 utilizamos algunos de estos modelos para generar melodías aleatorias. En el capítulo 3 discutimos algunos criterios para comparar modelos y los aplicamos a los aquí presentados.

2.3. Modelos para altura

Para simular las alturas de modo que la melodía resultante sea coherente con la tonalidad deseada, fijaremos algunas notas admitiendo sólo melodías que respeten dicha restricción. Entre las notas fijas, se eligen al azar las restantes utilizando una forma particular de cadenas de Markov que describimos a continuación.

2.3.1. Cadenas de Markov con restricciones

Sean $\{X_i\}_{i \in \mathbb{N}}$ una sucesión de variables aleatorias, $S = \{1, \ldots, K\}$ un conjunto finito que llamamos *espacio de estados*.[1] Diremos que $\{X_i\}$ es una *cadena de Markov* si verifica

$$\mathbf{P}(X_n = j \mid X_{n-1} = i, X_{n-2} = i_{n-2}, \ldots, X_0 = i_0) = \mathbf{P}(X_n = j \mid X_{n-1} = i),$$

[1] No es realmente necesario que sea finito, pero basta para nuestro caso de estudio.

para todo par de estados $i, j \in S$ e índice $n \in \mathbb{N}$. Si además la probabilidad $\mathbf{P}(X_n = j \mid X_{n-1} = i)$ sólo depende de i, j (no de n) decimos que la cadena es *homogénea*. Así, una cadena de Markov (homogénea) queda definida por una *matriz de transición* $P \in \mathcal{M}_{K\times K}$ y el vector de probabilidades del primer símbolo $\mu \in [0,1]^K$, conocido como *distribución inicial*.

Queremos ahora generar trayectorias finitas $x_0, x_1, \ldots, x_N$ a partir de la cadena de Markov definida por (P, μ), pero con ciertas *restricciones* en los distintos instantes $0, 1, \ldots, N$. Dichas retricciones podemos clasificarlas como

- **Restricciones unitarias:** son condiciones que refieren a un único instante $k \in \{0, 1, \ldots N\}$ y consisten en indicar cuáles son los estados que puede tomar la variable X_k. Llamamos $U_k \subset S$ al conjunto de los estados que pueden ser visitados en el instante k.
- **Restricciones binarias:** son condiciones que refieren a dos instantes consecutivos $k-1$ y k ($k \in \mathbb{N}$) e indican cuáles son las transiciones permitidas de X_{k-1} a X_k. Llamamos $B_k \subset S \times S$ al conjunto de tales transiciones.

Además buscamos que $\tilde{\mathbf{P}}$, probabilidad que la cadena con restricciones asigna a cada trayectoria sea proporcional a $\mathbf{P}$, probabilidad bajo la cadena homogénea original, es decir

$$\tilde{\mathbf{P}}(s) = \begin{cases} 0 & \text{si } s \notin T'_N, \\ \mathbf{P}(s \mid s \in T'_N) & \text{si } s \in T'_N, \end{cases} \tag{2.3}$$

donde T'_N es el conjunto de trayectorias de largo N que satisfacen las restricciones.

Las distribución de las secuencias $\{X_0, X_1, \ldots, X_N\}$ responde a una cadena de Markov no homogénea, cuyas matrices de transición $\{\tilde{M}_1, \ldots, \tilde{M}_N\}$ y distribución inicial $\tilde{\mu_0}$ necesitamos para la simulación. Para hallarlas utilizamos el algoritmo propuesto por Pachet en [PRB11], que describimos a continuación.

Observemos en primer lugar que las nuevas probabilidades deben definirse de modo tal que toda trayectoria que comience verificando las restricciones debe poder finalizarse. Para ello, cada vez que impongamos una restricción debemos propagarla del siguiente modo:

- **Fijación de estados:** si se quiere imponer una condición de la forma $U_k = \{a\}$,

hay que eliminar de U_{k+1} todos los estados a los que no se puede acceder desde a. De modo similar, hay que quitar de U_{k-1} los estados que no pueden ir hacia a.

- **Remoción de estados:** Cada vez que se quite un estado a del conjunto U_k, habrá que quitar de U_{k+1} todos los estados a los que sólo se puede acceder desde a. De U_{k-1} quitaremos los estados que sólo podían ir hacia a.

El proceso termina cuando en los pasos anteriores no hay más nada por hacer. Considerando las matrices de transición, el procedimiento consiste en construir una familia auxiliar de matrices $\{\mathbf{Z}^{(k)}\}_{\{k\in 0,\dots,N\}}$ que indica qué estados y transiciones están permitidos (generalmente no son estocásticas). Luego se renormalizan las matrices obtenidas de modo que sean estocásticas[1] y generen trayectorias con las probabilidades deseadas.

Construimos la familia $\{\mathbf{Z}^{(k)}\}_{k\in\{0,\dots N\}}$ del siguiente modo:

- **Inicialización.** Definimos $\mathbf{Z}^{(0)} = \mu$ y $\mathbf{Z}^{(k)} = \mathbf{P}$, para todo $k \in \{1, \dots, N\}$.
- **Remoción de estados.** Llamamos $z_{ij}^{(k)}$ a la entrada i, j de la matriz $\mathbf{Z}^{(k)}$. Para cada $j \notin U_k$, se establece $z_{ij}^{(k)} = 0$, para todo $i \in S$ (es decir, se lleva la j-ésima columna de la matriz a 0).
- **Remoción de transiciones.** Las transiciones prohibidas imponen ceros en las matrices del siguiente modo: para todo $i, j \in E,\ k \in \{1, \dots, N\}$ tales que $(i, j) \notin B_k$ se establece $z_{ij}^{(k)} = 0$.

A partir de las matrices $\mathbf{Z}^{(k)}$ se construyen las matrices de transición $\tilde{\mathbf{P}}_\mathbf{k}$ y la distribución inicial $\tilde{\mu}$, cuyas entradas se definen recursivamente

$$\begin{aligned}
\tilde{p}_{ij}^{(N)} &= \tfrac{z_{ij}^{(N)}}{\alpha_i^{(N)}}, & \alpha_i^{(N)} &= \sum_{l\in S} z_{il}^{(N)}, \\
\tilde{p}_{ij}^{(k)} &= \tfrac{\alpha_j^{(k+1)} z_{ij}^{(k)}}{\alpha_i^{(k)}}, & \alpha_i^{(k)} &= \sum_{l\in S} \alpha_l^{(k+1)} z_{il}^{(k)},\ k < N, \\
\tilde{\mu}_i &= \tfrac{\alpha_i^{(1)} z_i^{(0)}}{\alpha^{(0)}}, & \alpha^{(0)} &= \sum_{l\in S} \alpha_l^{(1)} z_l^{(0)}.
\end{aligned}$$

En caso de que $\alpha_i^{(k)} = 0$ impondremos $\tilde{p}_{ij}^{(k)} = 0$. Se verifica fácilmente que las probabilidades así definidas verifican la ecuación (2.3).

Con este modelo generamos las alturas de la forma que describimos a continuación.

[1] Abusando de la nomenclatura, admitiremos por estocásticas matrices que tengan filas de ceros.

2.4. Generación de melodías aleatorias

En esta sección presentamos una estrategia para la simulación de melodías sencillas con alturas y rítmica aleatorias procurando que éstas tengan cierto sentido, esto es, que respeten una estructura métrica y tonal definidas. Retomamos para ello lo propuesto e implementado en [Rum17] para la simulación de alturas a partir de una rítmica dada. Incorporamos además un método para simular ritmos, de modo que tenemos el siguiente esquema de generación:

- Simulamos una secuencia rítmica del largo deseado.
- Sobre la secuencia rítmica obtenida, se simulan las alturas correspondientes.

Dedicamos los siguientes apartados a discutir ambos procedimientos. Entretanto simplemente nótese que necesitamos estipular algunos elementos de antemano, como la duración (en cantidad de pulsos) de la secuencia deseada y ciertas alturas fijas que nos ayudarán a preservar la estructura tonal. Además definiremos modelos probabilísticos para simular el ritmo y, por separado, las alturas. Para nuestros ejemplos tomamos una melodía de referencia ("Arroz con leche") a partir de la cual simulamos otra que preserve ciertos rasgos de la original: la armonía por la cual se rige y su extensión (cantidad de compases).

Esta parte del trabajo es una extensión de lo realizado en [Rum17], donde se plantea un ejemplo similar pero con rítmica también fija. Tanto en aquel trabajo como en este las nuevas melodías se presentan como *variaciones*, en tanto preservan la armonía original.

2.4.1. Simulación de duraciones

Comenzamos simulando la rítmica de acuerdo a lo expresado en el capítulo anterior, es decir, considerando una grilla de pulsos sobre la cual se determinan los lugares de ataque. Utilizamos para nuestros ejemplos el modelo jerárquico propuesto (y preferido) por Temperley, al cual denominamos M_5, y el modelo jerárquico refinado (que notamos M_7) que proponemos como una mejora a éste.

Utilizamos las probabilidades de ataque correspondientes estimadas a partir de las obras en $\frac{2}{4}$ del conjunto de obras *Essen Folksong Collection*. En el capítulo 4 se describe detalladamente la elección y procesamiento de este y otros corpus. Por el momento, nos limitamos a considerar conocidos los parámetros de cada modelo.

Luego se generan las secuencias rítmicas, que representamos como secuencias de unos y ceros. Como nuestros ejemplos tienen unas pocas notas fijas preestablecidas, nuestras secuencias tendrán algunos valores fijos. El resto de los valores de la secuencia se obtienen del siguiente modo:

- Se simulan los ataques de posición métrica (en el compás) 0, es decir los de mayor nivel, de forma independiente y con probabilidad fija.
- Se simulan los ataques de posición métrica 2, correspondientes al segundo nivel de fuerza métrica. Las probabilidades en cada ataque dependen de su tipo de anclaje.
- Se simulan los ataques en posiciones métricas 1 y 3, correspondientes al nivel de menor fuerza métrica. Bajo M_5, las probabilidades dependen únicamente de su tipo de anclaje, mientras que bajo M_7 se tienen probabilidades distintas para cada una de las posiciones (las cuales también están condicionadas al tipo de anclaje).

En la figura 2.11 se ilustra el procedimiento presentando la generación paso a paso de la rítmica de una variación de Arroz con Leche.

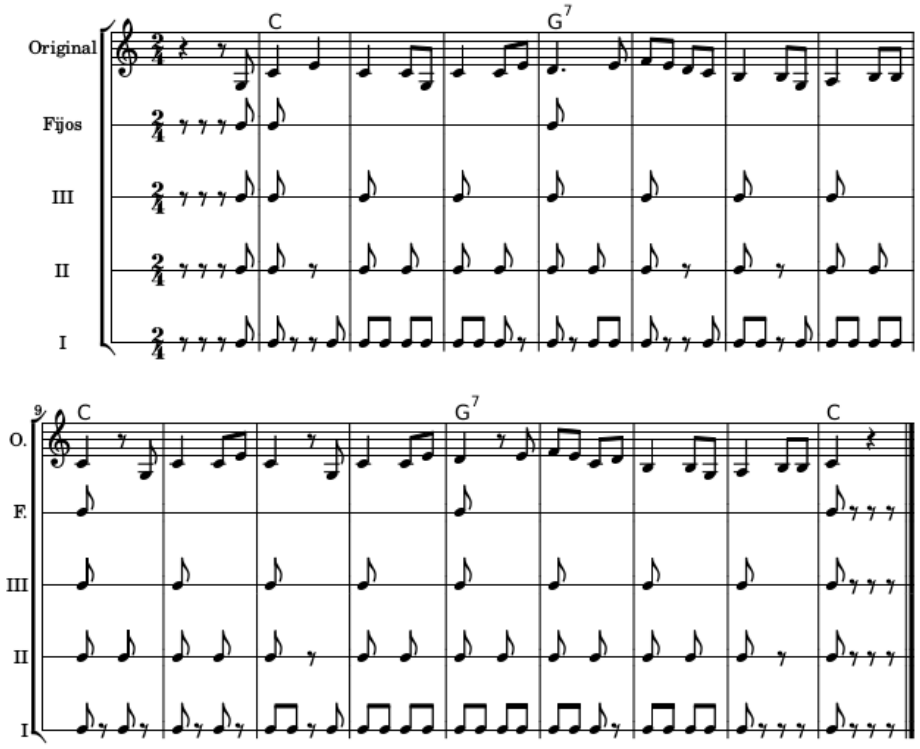

Figura 2.11: Melodía original de "Arroz con leche", indicando las posiciones fijas y la generación de un ejemplo nivel a nivel (indicados a la izquierda), utilizando el modelo M_5.

2.4.2. Simulación de alturas

Representamos las distintas alturas musicales como estados de una cadena de Markov con restricciones. Para ello asociamos a cada nota un estado (asumimos que nuestros estados son números enteros consecutivos) según algún criterio preestablecido. Algunas posibles elecciones son:

- Considerar un espacio de estados de 12 elementos, correspondientes a cada una de las clases de altura presentes en la escala cromática. La octava correspondiente se determina con algún criterio adicional, como ser la representante de la clase de altura más cercana a la altura que le precede y/o no salir de cierto rango de alturas (por ejemplo en el caso de estar simulando música vocal, tiene sentido acotar las alturas según el registro de dicha voz). En este caso las melodías podrán contener cualquier nota dentro de la escala cromática por lo que es muy importante una correcta definición de las matrices de transición y distribución inicial para preservar la estructura tonal. Una opción para ello es estimar los parámetros de una cadena homogéna auxiliar a partir de un corpus de obras en una misma tonalidad. Hay que definir además un criterio para establecer las restricciones.
- Establecer explícitamente las notas asociadas a cada estado (sin agrupar por clases de octava), eligiéndolas manualmente. Más aún, se pueden considerar distintos diccionarios a lo largo de la secuencia, por ejemplo, según la armonía subyacente. Este enfoque presenta algunas dificultades extra en la práctica ya que no es fácil detectar la armonía de forma automática cuando no se encuentra explícitamente anotada, con lo cual tampoco es fácil determinar el diccionario de notas a utilizar. Por otra parte, los lugares donde hay un cambio de acorde pueden ser utilizados como restricciones.

  En este caso buena parte de la estructura está dada por estos aspectos y la elección de la cadena homogénea sobre la cual se aplican las restricciones resulta menos crítica.

Una opción, poco interesante pero prudente en cuanto a la conducción melódica, es considerar un paseo al azar modificado para admitir transiciones de un estado en sí mismo. Es decir que las probabilidades de transición p_{ij} de la cadena homogénea original se definen como

$$p_{ij} = \begin{cases} p, & \text{si } j = i - 1, \\ q, & \text{si } j = i, \\ r, & \text{si } j = i + 1, \\ 0, & \text{en otro caso.} \end{cases} \tag{2.4}$$

Luego, se utiliza una cadena de Markov con restricciones construida a partir de una cadena homogénea con un estado inicial fijo, esto es, $\mathbf{P}(X_0 = 0) = 1$ y matriz de transición definida como en la ecuación (2.4).

En el ejemplo que consideramos usamos esta estrategia, con diccionario variable según la armonía. Una aplicación del caso que utiliza toda la escala cromática con estimación a partir de un corpus, así como una presentación más detallada de las cadenas de Markov con restricciones, pueden verse en [Rum17].

2.4.3. Ejemplos de variaciones aleatorias de una melodía dada

Siguiendo los procedimientos descritos en las secciones anteriores generamos variaciones aleatorias de Arroz con Leche. Disponemos inicialmente de la melodía, con su armonía explícitamente anotada. Consideramos pulsos de corchea y elegimos como notas fijas las ubicadas en cambios de acorde, como se ilustra en la figura 2.12.

Luego simulamos la secuencia rítmica de acuerdo a lo visto anteriormente, considerando ataques fijos en los lugares de las restricciones. Obtenemos una secuencia de unos y ceros con una rítmica asociada como la de la figura 2.11, sobre la cual generamos la melodía a través de un paseo al azar modificado con restricciones como el presentado en la sección anterior. Con él obtenemos una secuencia numérica que asociaremos a diferentes alturas según el acorde subyacente. Consideramos dos posibles diccionarios, uno para el acorde de G7 y otro para C, que utilizan únicamente notas del acorde.[1] En la figura 2.13 podemos ver ambos diccionarios.

Finalmente, se asignan las alturas simuladas a la rítmica previamente obtenida, obteniendo así una posible variación como la de la figura 2.14. De igual modo simulamos una variación de Arroz con Leche utilizando el modelo M_7 para el ritmo. Presentamos en la figura 2.15 el esquema completo de su construcción.

[1] Es posible utilizar otras estrategias menos restrictivas, como por ejemplo escalas que dependen del acorde.

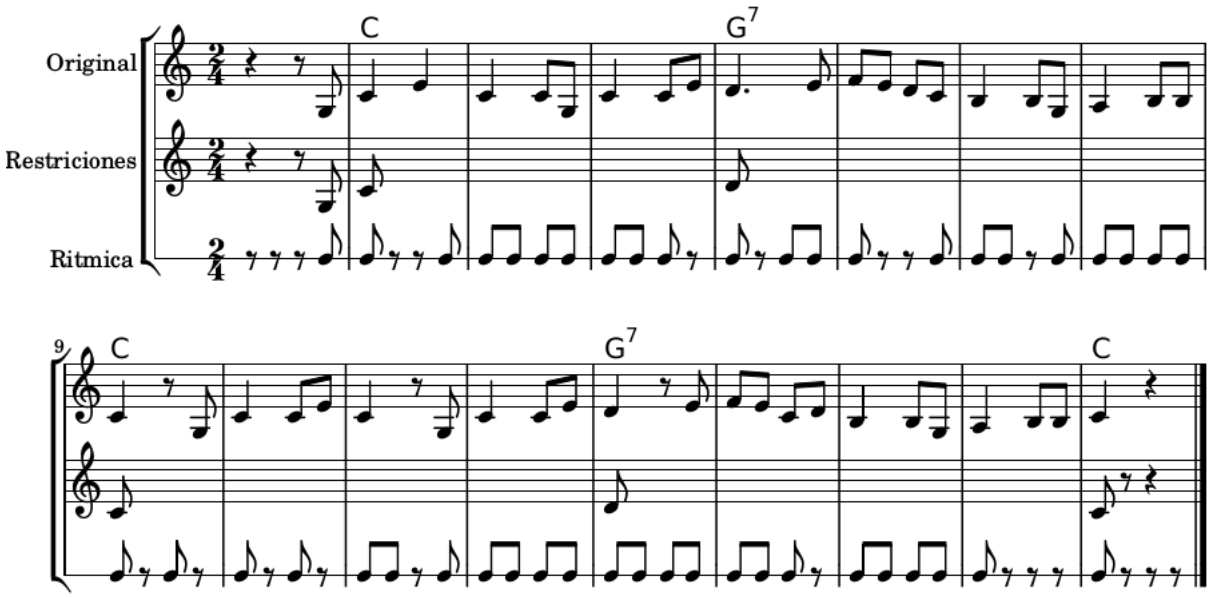

Figura 2.12: Melodía original de Arroz con leche y las notas elegidas como restricciones. Se agrega abajo la rítimica mostrada en 2.11, de la cual obtenemos la cantidad de notas a simular.

En ambos ejemplos si observamos la rítmica compás a compás podemos ver que aparecen algunos patrones que no se presentan en la melodía original, lo cual es razonable teniendo en cuenta que ésta no se utilizó en la estimación de las probabilidades de ataque. Mas no es nuestro objetivo discutir estadísticamente el ajuste de nuestros patrones rítmicos al verdadero Arroz con Leche, y nuestros ejemplos tienen por fin ilustrar el método utilizado para generarlos. Aquellas personas interesadas en obtener más variaciones pueden encontrar más ejemplos en [rep], así como el código para simularlas.

Es claro que nuestro esquema de simulación presenta varias limitaciones. Ofrece sin embargo una estrategia de simulación de melodías sencillas que se perciben bien estructuradas métrica y tonalmente (podríamos haber considerado otras estrategias que dieran lugar a melodías más o menos interesantes y más o menos estructuradas). Resulta difícil sin embargo definir cuándo nuestra melodía esta mejor adecuada a cierto estilo o corpus de referencia. En este punto consideramos necesario disponer de un criterio de comparación de modelos que podamos utilizar para evaluar las distintas posibles técnicas de simulación. Así, dedicamos el siguiente capítulo a presentar una estrategia de comparación de modelos que aplicamos sobre los modelos rítmicos ya estudiados.

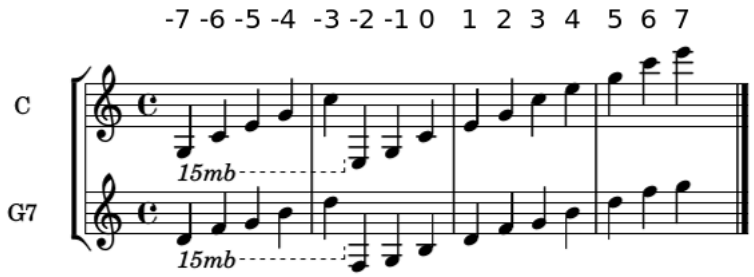

Figura 2.13: Alturas asignadas a los distintos números enteros (arriba) según los acordes de do (C) y sol7 (G7). Ambos diccionarios pueden extenderse aún más siguiendo el mismo criterio (notas del arpegio).

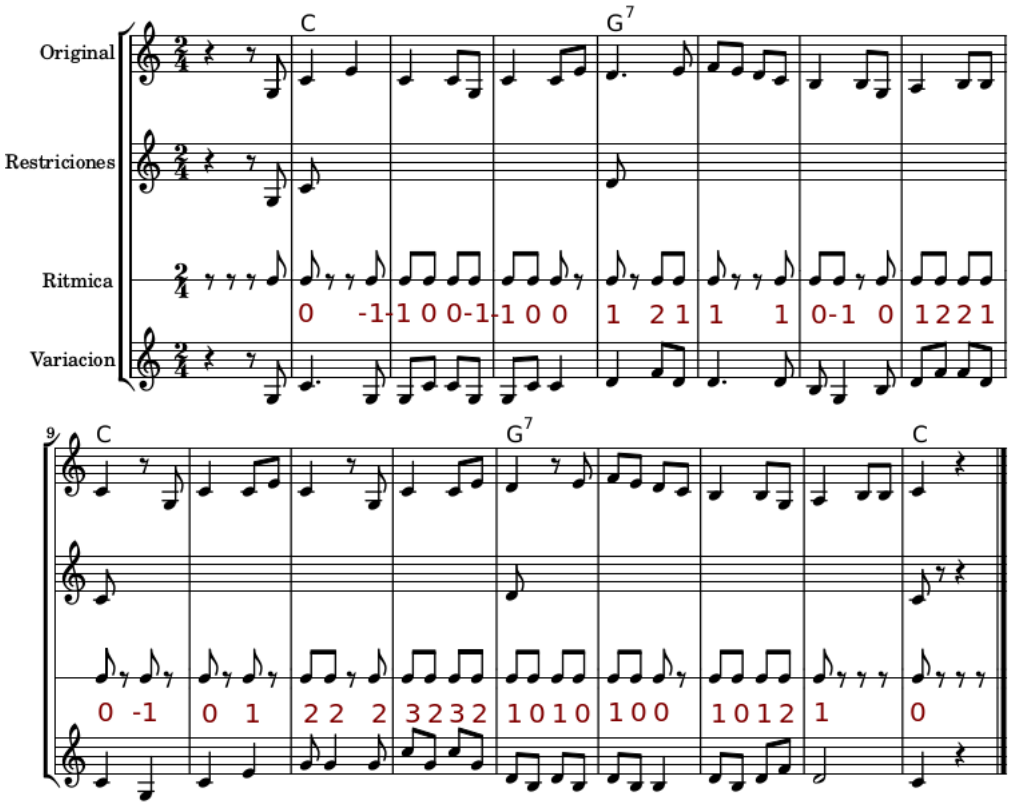

Figura 2.14: Esquema completo de la generación de una variación de Arroz con Leche, utilizando el modelo M_5. Se incluye la secuencia numérica obtenida mediante una cadena de Markov con restricciones. Escribimos el ejemplo simulado con duraciones completas (sin silencios).

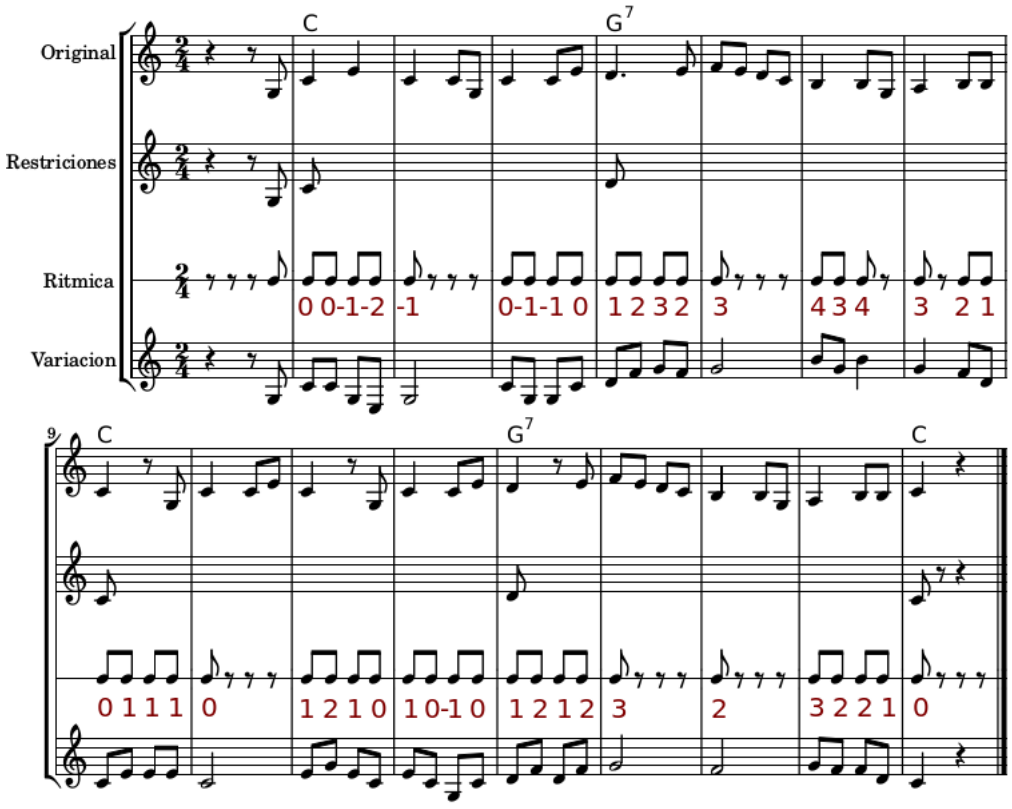

Figura 2.15: Esquema completo de la generación de una variación de Arroz con Leche, utilizando el modelo M_7.

Capítulo 3

Selección de modelos rítmicos

3.1. Comparando modelos con Crude Two-part MDL

3.1.1. Preliminares: Entropía y compresión

Consideremos un conjunto de datos D cuyos elementos toman valores en un conjunto finito (por ejemplo, $\{0, 1\}$) y supongamos que queremos representar D usando símbolos de otro conjunto, también finito. Queremos pues definir un *código* que nos permita convertir secuencias de símbolos como las de D en secuencias de símbolos en el nuevo diccionario. A partir de un modelo probabilístico podemos definir un código "eficiente" en el sentido de que asigne palabras más cortas a las secuencias más probables. No nos detendremos en detallar posibles estrategias de construcción de códigos (que pueden verse, por ejemplo, en [CT06]) ya que no nos interesa codificar explíctamente los datos. Simplemente observaremos que en un modelo paramétrico distintas elecciones de los parámetros dan lugar a códigos distintos, y que si lo que buscamos es una representación más breve de los datos debemos elegir parámetros que maximicen su probabilidad.

En este sentido, una posible estrategia para comparar el ajuste de los modelos a cierto conjunto de datos D puede ser considerar que el mejor modelo es aquel que logra la mayor compresión. Se entiende que comprimir eficazmente un conjunto de datos implica capturar las regularidades que lo caracterizan.

Por otra parte, Temperley propone un criterio para evaluar el ajuste de sus modelos frente a un corpus dado. Dicho criterio busca la menor entropía media "por obra" del corpus bajo la distribución de probabilidad asignada por el modelo, lo cual es similar a la

idea de compresión previamente planteada. A modo de introducción recordemos algunos resultados al respecto.

Definición 3.1.1. Sea X una variable aleatoria discreta con valores en un conjunto $\mathcal{X}$ y función de probabilidad p. Definimos su *entropía* como

$$H(X) := \mathbf{E} \log \left(\frac{1}{p(X)} \right) = - \sum_{x \in \mathcal{X}} p(x) \log(p(x)).$$

Observación. En la definición anterior, así como en el resto de este trabajo, utilizamos la notación log para referirnos al logaritmo en base 2.

Definición 3.1.2. Consideremos $X \in \mathcal{X} = \{x_1, \ldots, x_k\}$ una variable aleatoria discreta con probabilidades puntuales $\mathbf{p} = \{p_1, \ldots, p_k\}$. Además consideremos un código que representa cada elemento x_i de $\mathcal{X}$ usando $l(x_i)$ bits.

- Definimos el *largo esperado* del código como

  $$L := \mathbf{E} l(X) = \sum_{x_i \in \mathcal{X}} l(x_i) p_i,$$

  y diremos que un código es *óptimo* para $\mathcal{X}$ (con sus probabilidades $\mathbf{p}$) si minimiza L.
- Generalizando, si tenemos ahora una sucesión $\{X_i\}_{i \in \mathbb{N}}$ de variables aleatorias, definimos el largo medio de descripción por símbolo como

  $$L_n := \frac{1}{n} \mathbf{E} l(X_1, \ldots, X_n).$$

Notemos que utilizando un alfabeto $\mathcal{D}$ fijo no es posible construir un buen código utilizando únicamente palabras cortas.[1] Elegir representar algunos símbolos con palabras muy cortas obliga a usar palabras largas para representar otros. La desigualdad de Kraft nos da una cota para los largos de cada palabra.

Teorema 3.1.3 (Desigualdad de Kraft)**.** *Sea $\mathcal{S}$ un conjunto numerable de símbolos, que representamos utilizando un código instantáneo con alfabeto $\mathcal{D}$. Los correspondientes largos de palabra l_i verifican*

$$\sum_{i=1}^{\infty} |\mathcal{D}|^{-l_i} \leq 1,$$

[1] Nos referimos por buen código a uno instantáneo. Si bien no necesitaremos más detalles al respecto, éstos pueden verse en [CT06].

y recíprocamente, para toda familia de enteros positivos $\{l_i\}$ que verifique dicha desigualdad, existe un código instantáneo para $\mathcal{S}$ con palabras de largo l_i.

Por otra parte, la entropía se relaciona directamente con el largo medio de descripción de acuerdo a lo formulado en el Teorema de Codificación de Fuente [Shannon, 1948], el cual podemos expresar de la siguiente manera:

Teorema 3.1.4. *Consideremos una sucesión $\{X_i\}_{i\in\mathbb{N}}$ de variables aleatorias discretas y un código óptimo para ellas. Su largo medio de descripición por símbolo L_n^* verifica*

$$\frac{1}{n}H(X_1,\ldots,X_n) \leq L_n^* \leq \frac{1}{n}H(X_1,\ldots,X_n) + \frac{1}{n}.$$

En particular si las variables son independientes resulta

$$H(X_1) \leq L_n^* < H(X_1) + \frac{1}{n}$$

Por lo tanto tenemos que minimizar la entropía (o maximizar la probabilidad) es también comprimir el conjunto de datos. Sin embargo esta estrategia presenta el riesgo de *sobreajuste*, es decir, de considerar un modelo que describa bien los datos dados pero sea poco flexible a otros conjuntos de similares características. Esto suele ocurrir por el uso de modelos excesivamente complejos. Para evitar el sobreajuste Temperley penaliza aquellos modelos que tienen demasiados parámetros aún cuando logran comprimir más el corpus. Carece, sin embargo, de un criterio riguroso para determinar cuándo dicha mejora en la compresión no compensa el incremento en la complejidad del modelo.

3.1.2. Códigos de dos partes

Presentaremos ahora un criterio para comparar el ajuste de distintos modelos paramétricos a un conjunto de datos dado, penalizando el sobreajuste. Consideremos una familia de modelos paramétricos a comparar y un conjunto $D = \{x_0,\ldots,x_{N-1}\}$ que queremos codificar y cuyos elementos, por simplicidad, supondremos en $\{0,1\}$ (aunque esto no es realmente necesario). Para que el conjunto original pueda reconstruirse a partir de dicha codificación debemos representar los datos, pero también el modelo empleado para la codificación y en particular sus parámetros. Bajo esta representación, se busca el *mínimo largo de descripción* (MDL, por sus siglas en inglés).

Ejemplo 3.1.5. Sean D una secuencia rítmica (entiéndase una secuencia de unos y ceros como las anteriormente definidas), y llamaremos $\{M_\alpha\}_{\alpha=1,\dots,6}$ a los 6 modelos de Temperley. Si queremos representar D a partir de dichos modelos debemos codificar:

- Una etiqueta que indique cuál de las 6 familias M_α se está utilizando (notemos que esto es simplemente representar un número entre 1 y 6).
- Una vez conocida la familia, el valor de los parámetros a emplear.
- El conjunto D, a partir del modelo considerado.

Nótese que transmitiendo estas tres cosas un receptor que conozca el criterio usado para codificar podría reconstruir cualquier secuencia rítmica que le enviemos. Estamos transmitiendo tanto los datos como el modelo utilizado para codificarlo. Dado que lo que nos interesa son sólo los largos de descripción resultantes, omitiremos la codificación de la etiqueta que indica el código entendiendo que su mínimo largo de descripción es pequeño y con poca variación entre modelos.

En general se procurará minimizar, pues, el largo de descripción total, entendido como el largo de descripción de los parámetros del modelo más el largo de descripción de los datos codificados usando dichos parámetros. Así, dado un modelo M y un conjunto de datos D su largo de descripción total es:

$$L_{total}(D, M) = L_{C_1}(M) + L_{C_2}(D \mid M), \tag{3.1}$$

donde C_1 es el código utilizado para escribir los parámetros del modelo, y C_2 el utilizado para los datos. Discutamos brevemente el aporte de ambos términos al largo total de descripción.

1. *El código* C_1. Obsérvese que utilizamos este código para representar parámetros reales (en nuestro caso particular, probabilidades). Claramente no podemos representar todos los números en $[0, 1]$ con palabras finitas, con lo cual debemos en realidad truncar los parámetros a representar según cierta precisión finita previamente establecida. La elección de dicha precisión no es trivial y afecta a $L_{C_1}(M)$, en tanto mayor precisión implica una mayor cantidad de valores posibles a representar y, en consecuencia, palabras más largas.
   Sin entrar en detalles de la construcción de un código universal para los números reales (para más detalles al respecto ver [Gru07]), observaremos que si tomamos

una grilla que nos permita representar d valores, el largo medio de descripción por símbolo resulta $\log(d)$. Por lo tanto, si M tiene k parámetros, tenemos $L_{C_1}(M) = k\log(d)$. En caso de que d no esté establecido de antemano (por ejemplo, si lo definimos en función de D), habrá que considerar un $\log(d)$ extra que es el costo de transmitir el propio d.

2. *El código C_2.* Una vez establecidos los parámetros θ_i, con $i \in \{1, \ldots, k\}$, queda definida la probabilidad de cualquier secuencia. Consideremos $\mathbf{P}_\theta : \{0,1\}^N \longrightarrow [0,1]$ la función de probabilidad para las secuencias de N elementos. La desigualdad de Kraft, nos garantiza que podemos definir un código (instantáneo) que asigne a cada secuencia $s \in \{0,1\}^N$ una palabra de código de largo

$$l(s) = \lceil -\log \mathbf{P}_\theta(s) \rceil.$$

Esto define el largo necesario para tranmitir las secuencias, mas no explicitamos el código utilizado. El modelo utilizado para codificar y sus parámetros se asumen ya conocidos.

Notemos que en este contexto es de esperar que N sea grande, ya que será el tamaño del conjunto de datos. En consecuencia los valores de $-\log \mathbf{P}_\theta(s)$ serán usualmente muy grandes. Por otra parte, si considerásemos $l(s) = -\log \mathbf{P}_\theta(s)$ (sin redondear) tenemos

$$H(X_0, \ldots, X_{N-1}) = \sum_{s \in \{0,1\}^N} -\mathbf{P}_\theta(s) \log \mathbf{P}_\theta(s) = \sum_{s \in \{0,1\}^N} \mathbf{P}_\theta(s) l(s) = L,$$

que es el menor largo medio de descripción posible.

Es decir que tenemos una estrategia para construir un código "casi" óptimo a partir de los parámetros dados, cualesquiera estos sean. Luego, los elegimos de modo que minimicen el largo de descripción de nuestros datos. Es decir

$$\hat{\theta} = \underset{\theta \in \Theta_k \subset [0,1]^k}{\arg\min} -\log \mathbf{P}_\theta(D),$$

siendo Θ_k el espacio de los parámetros posibles. Como ya se observó previamente, dicho $\hat{\theta}$ es efectivamente el estimador por máxima verosimilitud de θ. Se desprende así el segundo término de la ecuación (3.1):

$$L_{C_2}(D \mid M) = -\log \mathbf{P}_{\hat{\theta}}(D).$$

En esta primer aproximación consideramos que efectivamente utilizamos $\hat{\theta}$ para construir C_2.

Tenemos así una fórmula para el largo de descripción, que denominamos *Crude Two-part MDL* de acuerdo a la nomenclatura definida por Grunwald en [Gru07].

$$L_{total}(D, M, d) = k \log(d) + log(d) - \log \mathbf{P}_M(D), \tag{3.2}$$

siendo k la cantidad de parámetros del modelo M. Obsérvese que se incorporó la variable d (precisión) a la ecuación y que el segundo término de la suma corresponde al costo de transmitir d (en los casos que corresponda). En general abusaremos de notación y utilizaremos indistintamente el nombre del modelo o el vector de parámetros θ subyacente (cuando la familia de modelos se sobreentienda).

Elección de d. Se desprende de la ecuación anterior que, independientemente del modelo considerado, el largo total de descripción es creciente en d. A mayor valor de d, mayor es el costo de codificar cada parámetro y por lo tanto mayor es la penalización por tomar modelos más complejos. Por otra parte, d representa la precisión a emplear al codificar los parámetros estimados por máxima verosimilitud, por lo que es razonable suponer que la precisión elegida dependerá de la calidad de la estimación (es decir, el tamaño del conjunto de datos).

De acuerdo a lo propuesto por Rissanen en [Ris78], [Ris83] y [Ris86] consideramos $d = \sqrt{N}$ se obtiene así el siguiente largo de descripción:

$$L_{total}(D, M) = \frac{k}{2} \log(N) + \cancel{\frac{1}{2} \log(N)} - \log \mathbf{P}_M(D). \tag{3.3}$$

El segundo término se muestra cancelado porque podría omitirse bajo este criterio de elección de la precisión si asumimos que el tamaño del conjunto D, así como la elección de d en función de éste, son conocidos de antemano por el receptor. Notemos que, esto resulta además muy similar a BIC (Bayesian Information Criterion)[Schwartz, 1978], que se define como

$$BIC(M, D) = \log\left(\mathbf{P}_{\hat{\theta}}(D)\right) - 2k \log(N),$$

siendo M una familia paramétrica de modelos con k parámetros, $\hat{\theta}$ el estimador máximo verosimil de sus parámetros, D el conjunto de datos y N su tamaño. Si en la ecuación (3.3) consideramos los parámetros estimados por máxima verosimilitud y omitimos el término tachado, tenemos que maximizar BIC es equivalente a minimizar nuestro largo de descripción.

3.1.3. Aplicación de Crude Two-Part a los modelos de Temperley

Utilizando Crude Two-part MDL podemos comparar los modelos para ritmo presentados sobre conjuntos de datos D dados. Retomando los cálculos de la sección 2.2.3 para el compás de $\frac{4}{4}$ subdividido en corcheas, hallamos los largos de descripción asociados a cada modelo. En primera instancia consideramos $d = \sqrt{N}$ y parámetros estimados por máxima verosimilitud. Así,

$$L_{total}(D, M) = \frac{k}{2}\log(N) + \frac{1}{2}\log(N) - \log \mathbf{P}_M(D).$$

$\mathbf{M_1}$ Este modelo tiene únicamente un parámetro ($k = 1$) por lo que $L_{C_1}(M_1) = \log(\sqrt{N}) + \log(\sqrt{N}) = \log(N)$. Recordemos que la probabilidad de los datos es

$$\mathbf{P}_{\mathbf{M_1}}(D \mid p) = p^{\sum_{i=0}^{N-1} x_i}(1-p)^{N-\sum_{i=0}^{N-1} x_i}.$$

Tenemos pues

$$L_{C_2}(D \mid M_1) = -\sum_{i=0}^{N-1}[x_i \log(\hat{p})) + (1 - x_i)\log(1 - \hat{p})],$$

resultando el largo de descripción total

$$L_{total}(D, M_1) = \log(N) - \left[\sum_{i=0}^{N-1} x_i \log(\hat{p})) + \sum_{i=0}^{N-1}(1 - x_i)\log(1 - \hat{p})\right].$$

Como $\hat{p} = \frac{1}{N}\sum x_i$, tenemos la expresión alternativa

$$L_{total}(D, M_1) = -N(\hat{p}\log(\hat{p}) + (1 - \hat{p})\log(1 - \hat{p})) + \log(N)$$

$\mathbf{M_2}$ En este caso se tienen $k = T$ parámetros (además de la precisión d), y considerando la probabilidad de D calculada en 2.2.3 se tiene

$$L_{total}(D, M_2) = \frac{T+1}{2}\log(N) - \sum_{j=0}^{T}\sum_{i=1}^{i=M} \mathbb{1}_{\{y_i=j\}} \log \hat{\theta}_j.$$

O bien, utilizando la expresión anteriormente hallada para $\hat{\theta}$ y observando que, de la ecuación (2.2) se desprende

$$M + 1 = N\hat{p},$$

tenemos

$$L_{total}(D, M_2) = \frac{T+1}{2}\log(N) - N\hat{p}\sum_{j=0}^{T}\hat{\theta}_j \log(\hat{\theta}_j).$$

M_3 El largo de descripción para este modelo de $k = 4$ parámetros resulta

$$\begin{aligned} L_{total}(D, M_3) =& \frac{5}{2}\log(N) \\ & - \sum_{i=\dot{8}}(x_i \log(\hat{\theta}_4) + (1-x_i)\log(1-\hat{\theta}_4)) \\ & - \sum_{i\equiv 4 \pmod 8}(x_i \log(\hat{\theta}_3) + (1-x_i)\log(1-\hat{\theta}_3)) \\ & - \sum_{i\equiv 2 \pmod 4}(x_i \log(\hat{\theta}_2) + (1-x_i)\log(1-\hat{\theta}_2)) \\ & - \sum_{i\equiv 1 \pmod 2}(x_i \log(\hat{\theta}_1) + (1-x_i)\log(1-\hat{\theta}_1)). \end{aligned}$$

Si definimos $N_1 = \frac{N}{2}$, $N_2 = \frac{N}{4}$ y $N_3 = N_4 = \frac{N}{8}$, resulta

$$L_{total}(D, M_3) = -\sum_{j=1}^{j=4} N_j(\hat{\theta}_j \log(\hat{\theta}_j) + (1-\hat{\theta}_j)\log(1-\hat{\theta}_j)) + \frac{5}{2}\log(N).$$

M_4 De modo similar al modelo anterior se tiene

$$L_{total}(D, M_4) = \frac{9}{2}\log(N) - \sum_{j=0}^{j=7}\left[\sum_{i\equiv j \pmod 8}(x_i \log(\hat{\theta}_j) + (1-x_i)\log(1-\hat{\theta}_j))\right],$$

o bien

$$L_{total}(D, M_4) = \frac{9}{2}\log(N) - \frac{N}{8}\sum_{j=1}^{j=8}(\hat{\theta}_j \log(\hat{\theta}_j) + (1-\hat{\theta}_j)\log(1-\hat{\theta}_j)).$$

M_5 Utilizando la notación definida en los cálculos de probabilidades para este modelo obtenemos

$$\begin{aligned} L_{total}(D, M_5) =& 7\log(N) - n_{IV}\log\hat{\theta}_{IV} - (\frac{N}{8} - n_{IV})\log(1-\hat{\theta}_{IV}) \\ & - \sum_{k\in\{I,II,III\}}\sum_{i,j\in\{0,1\}}\left[n_{k_{ij}}\log(\hat{\theta}_{k_{ij}}) + n^0_{k_{ij}}\log(1-\hat{\theta}_{k_{ij}})\right]. \end{aligned}$$

$\mathbf{M_6}$ El largo total de descripción según el modelo M_6 es

$$L_{total}(D, M_6) = \frac{57}{2}\log(N) - \sum_{i,j\in\{0,\ldots,7\}} n_{ij}\log\hat{p}_{ij},$$

con n_{ij} y p_{ij} definidos en la sección 2.2.3.

$\mathbf{M_7}$ Bajo M_7 con la notación definida en el capítulo anterior los datos tienen largo de descripción

$$\begin{aligned} L_{C_2}(D, M_7) = &- n_{IV}\log\hat{\theta}_{IV} - \left(\frac{N}{8} - n_{IV}\right)\log(1-\hat{\theta}_{IV}) \\ &- \sum_{\substack{k\in\{1,3,5,7,II,III\} \\ i,j\in\{0,1\}}} \left[n_{k_{ij}}\log(\hat{\theta}_{k_{ij}}) + n^0_{k_{ij}}\log(1-\hat{\theta}_{k_{ij}})\right], \end{aligned}$$

Considerando además el largo de descripción del modelo se tiene

$$\begin{aligned} L_{total}(D, M_7) = &13\log(N) - n_{IV}\log\hat{\theta}_{IV} - \left(\frac{N}{8} - n_{IV}\right)\log(1-\hat{\theta}_{IV}) \\ &- \sum_{\substack{k\in\{1,3,5,7,II,III\} \\ i,j\in\{0,1\}}} \left[n_{k_{ij}}\log(\hat{\theta}_{k_{ij}}) + n^0_{k_{ij}}\log(1-\hat{\theta}_{k_{ij}})\right]. \end{aligned}$$

$\mathbf{M}_{total}$ Para el modelo jerárquico más refinado tenemos

$$\begin{aligned} L_{C_2}(D, M_{total}) = &- n_0\log\hat{\theta}_0 - \left(\frac{N}{8} - n_0\right)\log(1-\hat{\theta}_0) \\ &- \sum_{\substack{k\in\{1,\ldots,7\} \\ i,j\in\{0,1\}}} \left[n_{k_{ij}}\log(\hat{\theta}_{k_{ij}}) + n^0_{k_{ij}}\log(1-\hat{\theta}_{k_{ij}})\right], \end{aligned}$$

de lo cual se desprende el largo de descripción total

$$\begin{aligned} L_{total}(D, M_{total}) = &15\log(N) - n_0\log\hat{\theta}_0 - \left(\frac{N}{8} - n_0\right)\log(1-\hat{\theta}_0) \\ &- \sum_{\substack{k\in\{1,\ldots,7\} \\ i,j\in\{0,1\}}} \left[n_{k_{ij}}\log(\hat{\theta}_{k_{ij}}) + n^0_{k_{ij}}\log(1-\hat{\theta}_{k_{ij}})\right]. \end{aligned}$$

En todos los casos el largo de descripción total depende del tamaño del conjunto de datos. Más aún, los largos L_{C_2} crecen linealmente con N mientras que los largos L_{C_1} lo

hacen logarítmicamente. En consecuencia, para conjuntos de datos muy grandes puede ser razonable tomar modelos complejos si esto realmente disminuye $L_{C_2}(D \mid M)$. Esto se observará empíricamente en el próximo capítulo, aplicando estos modelos a conjuntos de datos reales.

Por otra parte, esta estrategia supone que representamos el estimador máximo verosímil con precisión finita y estipulada con cierta arbitrariedad. Sin embargo, a la hora de codificar los datos, asumimos que podemos utilizar el verdadero estimador por máxima verosimilitud y no su representación truncada. De este modo, si efectivamente estuviésemos transmitiendo los datos, nuestro receptor no tendría forma de saber cuáles son los parámetros que *realmente* se usaron para definir el código C_2 y no podría decodificar el mensaje. En la siguiente sección proponemos una modificación a Crude Two-part MDL que sí refleje una estrategia real de transmisión de datos.

3.2. Una variante de Crude Two-part MDL

3.2.1. El espacio de parámetros

Al igual que en Crude two-part MDL comparamos el ajuste de nuestros modelos al conjunto de datos a través del largo de descripción total, buscando el modelo que nos permite codificar los datos (y el propio modelo) con mayor compresión. A grandes rasgos mantenemos los dos pasos principales:

- Se codifican los parámetros del modelo, con precisión finita d.
- Se codifican los datos según el modelo dado y los parámetros elegidos en el paso anterior.

A fin de que el supuesto receptor pueda decodificar el mensaje, debemos utilizar realmente los parámetros trasmitidos en el primer paso para codificar los datos. Veamos cómo afecta este cambio al cálculo del largo total de descripción.

Sea $\theta \in \Theta_k \subset [0,1]^k$ nuestro vector de parámetros. Como no podemos definir un código para todo el espacio Θ_k consideramos, dada una precisión d, el conjunto de parámetros

$$\Theta_k^d := \left\{ \theta \in \Theta_k : \theta_j = \frac{j}{d}, j \in \{0, \ldots, d\} \right\}.$$

Observemos que Θ_k^d es una grilla discreta de puntos dentro de Θ_k, como se puede apreciar en las figuras 3.1 y 3.2. El primer caso representa vectores en $\mathbb{R}^2$ de parámetros sin mayores restricciones, mientras que el segundo corresponde a vectores de probabilidad en $\mathbb{R}^3$ (que pueden proyectarse en $\mathbb{R}^2$ como indica la figura).

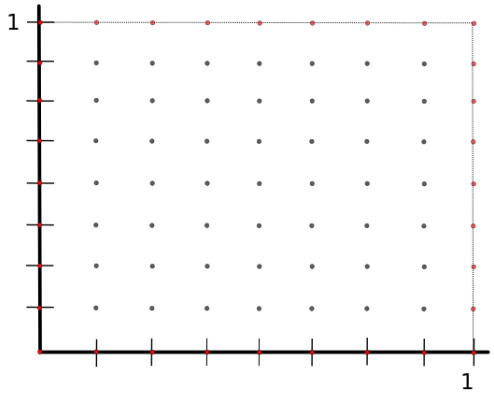

Figura 3.1: Los puntos representan los posibles valores de θ_d, con $k = 2$, $\Theta_k = [0,1]^2$ y $d = 8$. En la práctica evitaremos los puntos con coordenadas 0 o 1 (rojo), a no ser que dicha coordenada sea también 0 o 1 en $\hat{\theta}$.

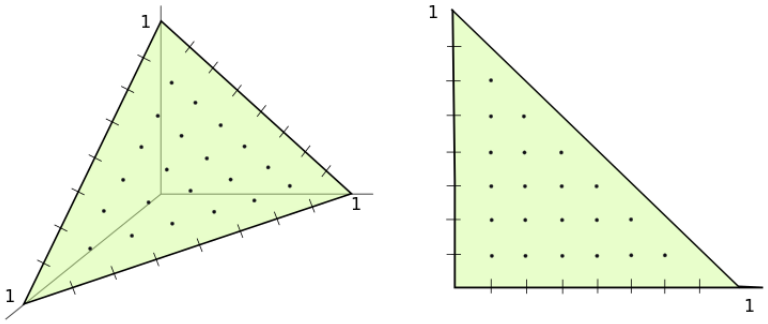

Figura 3.2: Posibles valores de θ_d cuando θ es un vector de probabilidad en $\mathbb{R}^3$, es decir, $k = 3$ y $\Theta_k = \{(x, y, z) \in [0,1]^3 : x + y + z = 1\}$. Nuevamente tomamos $d = 8$.

Por simplicidad para futuros cálculos, asumiremos que d es potencia de 2 y una vez más buscaremos minimizar

$$L_{total}(D, M, d) = k \log(d) + log(d) - \log \mathbf{P}_M(D). \tag{3.4}$$

En este caso, sin embargo, minimizar el último término no es trivial, ya que solo podemos considerar parámetros en Θ_k^d y el estimador máximo verosímil de θ podría no estar en dicho conjunto. Debemos encontrar pues, un nuevo estimador $\tilde{\theta_d}$ que maximice la verosimilitud restringida a Θ_k^d, es decir

$$\tilde{\theta_d} = \arg\min_{\theta_d \in \Theta_k^d} -\log \mathbf{P}_{\theta_d}(D). \tag{3.5}$$

Por lo tanto, tenemos dos variables sobre las cuales optimizar: Para d fijo debemos hallar el vector $\tilde{\theta_d}$ definido en la ecuación (3.5), y por otra parte también debemos buscar el valor de d que minimice el largo total de descripción. A diferencia de lo ocurrido al utilizar crude two-part MDL, en este caso el largo de descripción no tiene por qué ser monótono en d. Más aún, al incrementar d se verán afectados todos los términos de la ecuación (3.4) del siguiente modo:

- El largo de descripción del modelo, $k\log(d) + log(d)$, es creciente en d, al igual que en la sección anterior.
- El largo de descripción de los datos según el modelo, $-\log \mathbf{P}_{\theta_d}(D)$ es decreciente en d ya que consideramos d potencia de 2 y en consecuencia

  $$\Theta_k^{d_1} \subset \Theta_k^{d_2}, \text{ si } d_1 < d_2.$$

  Luego $-\log \mathbf{P}_{\tilde{\theta}_{d_2}}(D) \leq -\log \mathbf{P}_{\tilde{\theta}_{d_1}}(D)$. Intuitivamente, esto indica que tomando mayor precisión en la representación de los parámetros (y agrandando el conjunto de vectores posibles) podemos obtener mejores representaciones del conjunto de datos. Incluso podemos llegar a encontrar el verdadero estimador por máxima verosimilitud, a partir del cual $-\log \mathbf{P}_{\tilde{\theta}_d}$ se mantendrá constante.

3.2.2. Largo total de descripción según d

En primer lugar buscamos una estrategia para hallar $\tilde{\theta_d}$. Dado que el dominio sobre el cual estamos minimizando es finito, una primera (y generalmente inviable en la práctica) forma de hacerlo es calcular explícitamente $-\log \mathbf{P}_{\theta_d}(D)$ para todo $\theta_d \in \Theta_k^d$. Sin embargo el tamaño de Θ_k^d es del orden de d^k y habría que realizar el cálculo para valores de d muy grandes. Usamos entonces algunas estrategias más eficientes, dependiendo de la situación.

Fijado el conjunto de datos D y un modelo con k parámetros $\theta \in \Theta_k$, definamos

$g_D : \Theta_k \longrightarrow \mathbb{R}$ como

$$g_D(\theta) = -\log \mathbf{P}_\theta(D),$$

que es la función que queremos minimizar, en Θ_k^d.

Observación. En los modelos de Temperley las funciones g_D asociadas a cada uno de los 6 modelos son convexas. Para probarlo notemos que, más en general, esto vale si D puede escribirse como una secuencia $o_1, \ldots, o_L$ de observaciones de una cadena de Markov con espacio de estados E finito y probabilidad de transición de i a j p_{ij}. En tal caso

$$g_D(\theta) = -\log \mathbf{P}(D) = -\sum_{i,j \in E} n_i \hat{p}_{ij} \log(\theta_{ij}),$$

siendo $n_i = \sum_{k=1}^{L-1} \mathbb{1}_{\{o_k = i\}}$ la cantidad de transiciones desde i y $\hat{p}_{ij}$ el estimador por máxima verosimilitud de p_{ij}. Luego, dados $\theta^{(1)}, \theta^{(2)} \in \Theta_k$ y $t \in [0,1]$ se tiene

$$\begin{aligned} g_D(t\theta^{(1)} + (1-t)\theta^{(2)}) &= -\sum_{i,j \in E} n_i \hat{p}_{ij} \log(t\theta_{ij}^{(1)} + (1-t)\theta_{ij}^{(2)}) \\ &\overset{(1)}{\leq} -\sum_{i,j \in E} n_i \hat{p}_{ij} \left[t \log(\theta_{ij}^{(1)}) + (1-t) \log(\theta_{ij}^{(2)}) \right] \\ &= t g_D(\theta^{(1)}) + (1-t) g_D(\theta^{(2)}), \end{aligned}$$

donde la desigualdad (1) se desprende de la concavidad de la función log.

En particular el resultado vale para variables i.i.d., lo cual comprende a todos los modelos de Temperley excepto M_5. Para dicho modelo se tiene que los compases forman una cadena de Markov, es decir, para $i \in \{1, \ldots, \frac{N}{8}\}$ consideramos $o_i = (x_{8(i-1)}, \ldots, x_{8i-1})$ y se cumplen las hipótesis de lo recién demostrado.

La convexidad de g_D nos garantiza la existencia de un mínimo y resulta útil para calcularlo. Distiguimos dos casos:

Caso 1. La función g_D puede descomponerse como $g_D(\theta) = \sum_{i=1}^{i=k} h_i(\theta_i)$, con $\Theta_k = (0,1)^k$.[1] En este caso, que contempla todos los modelos de Temperley excepto M_2 y M_6, podemos simplemente minimizar g_D coordenada a coordenada. De este modo no sólo se

[1] Excluimos la posibilidad de que alguna coordenada sea 0 o 1 para evitar problemas con la función log. En la práctica, además, son valores que no incrementan el largo de descripción de D.

reduce la cantidad de puntos a evaluar sino que en caso de que las funciones h_i sean convexas (como en el caso de los modelos de Temperley) tenemos

$$\tilde{\theta}_{d_i} \in \left\{ \frac{b_i}{d}, \frac{b_i + 1}{d} \right\},$$

siendo b_i tal que $\frac{b_i}{d} \leq \hat{\theta}_i < \frac{b_i+1}{d}$ (mantendremos la notación $\hat{\theta}$ para referirnos al estimador por máxima verosimilitud). Este resultado se ilustra en la figura 3.3.

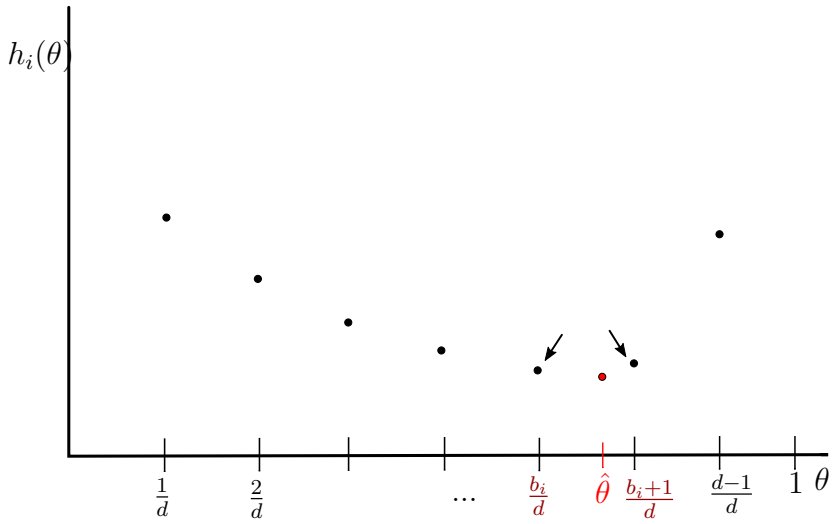

Figura 3.3: Si h_i es convexa, tenemos que $\tilde{\theta}_{di}$ es necesariamente uno de sus dos vecinos más cercanos en la grilla.

Así, como podemos calcular analíticamente $\hat{\theta}$, sólo tenemos que evaluar $2k$ puntos.

Caso 2. La función g_D es tal que no podemos separarla coordenada a coordenada, por ejemplo, porque θ es un vector de probabilidad (con lo cual el dominio no es de la forma $(0,1)^k$). Este caso comprende los modelos M_2 y M_6 de Temperley. Para hallar el mínimo en este caso utilizaremos el Lema del Epígrafo [BV04]:

Lema 3.2.1. *Sean $A \subset \mathbb{R}^k$ un conjunto convexo, $g : A \longrightarrow \mathbb{R}$, una función y $\mathcal{L}_\alpha$ los conjuntos tales que*

$$\mathcal{L}_\alpha = \{x \in A : g(x) \leq \alpha\}.$$

Si g es convexa, entonces los conjuntos $\mathcal{L}_\alpha$ son convexos.

Demostración. Sean x, $y \in \mathcal{L}_\alpha$. Como g es convexa, para todo $t \in [0,1]$ se verifica

$$\begin{aligned} g(tx + (1-t)y) &\leq tg(x) + (1-t)g(y) \\ &\leq t\,\text{máx}\{g(x), g(y)\} + (1-t)\,\text{máx}\{g(x), g(y)\} \\ &\leq \text{máx}\{g(x), g(y)\} \leq \alpha, \end{aligned}$$

pues $g(x) \leq \alpha$ y $g(y) \leq \alpha$. Por lo tanto $tx + (1-t)y \in \mathcal{L}_\alpha$ y en conclusión $\mathcal{L}_\alpha$ es convexo. □

Consideremos un punto a en el dominio A de una función diferenciable y convexa. Como consecuencia del lema, si consideramos su conjunto de nivel $\mathcal{C}_a$ y el hiperplano tangente a $\mathcal{C}_a$ por a, tenemos que $\mathcal{L}_a$ queda contenido en uno de los semiespacios que éste delimita. Esto nos da una estrategia para hallar $\tilde{\theta_d}$.

- Supongamos que $\Theta_k \subset (0,1)^k$ y por lo tanto $\hat{\theta}$ tiene todas sus coordenadas no nulas. Además asumamos que $\hat{\theta}$ no pertenece a Θ_k^d (en caso contrario, es el $\tilde{\theta_d}$ buscado). Definimos el conjunto de sus vecinos en la grilla de nivel l como

  $$\mathcal{N}_l = \left\{ \theta \in \Theta_k^d : \theta_i \in \left\{ \frac{j_i - l + 1}{d}, \ldots, \frac{j_i + l}{d} \right\}, i \in \{1, \ldots, k\} \right\},$$

  donde $j_i < d$ es el entero positivo tal que $\frac{j_i}{d} \leq \theta_i < \frac{j_i+1}{d}$. Notemos que $\mathcal{N}_1$ es un conjunto similar al definido coordenada a coordenada para el caso 1, con las restricciones adicionales que la forma de Θ_k^d imponga. En particular, puede ser vacío.
- Sea l_0 el menor l tal que $\mathcal{N}_l$ es no vacío. Para cada punto $x \in \mathcal{N}_{l_0}$ consideramos el hiperplano tangente al conjunto de nivel de g_D por x y llamamos Q_x al semiespacio (con borde) delimitado por dicho plano que contiene a $\hat{\theta}$. Se verifica así que $\tilde{\theta}_d \in Q_x$, para todo $x \in \mathcal{N}_{l_0}$.
- Definimos

  $$Q^d = \left(\bigcap_{x \in \mathcal{N}_{l_0}} Q_x \right) \cap \Theta_k^d.$$

  Obsérvese que Q^d es no vacío (contiene al menos un elemento de $\mathcal{N}_{l_0}$) y en particular contiene a $\tilde{\theta}_d$.
- Finalmente, se calcula g_D para todo elemento en Q^d, encontrando el mínimo buscado.

En la figura 3.4 se ejemplifica esta construcción para $k = 3$. Los puntos en Q^d son

los representados con relleno, mientras que los puntos "huecos" son aquellos que nos ahorramos visitar con este procedimiento.

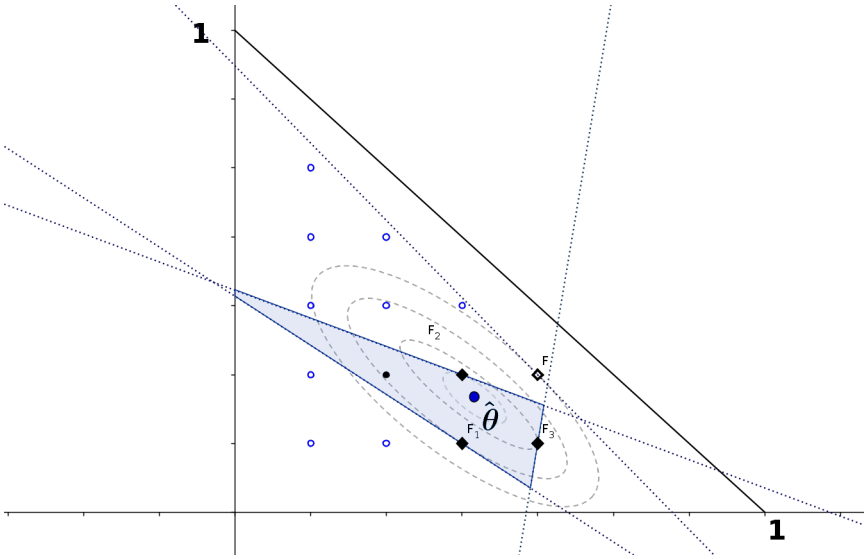

Figura 3.4: Ejemplo de construcción del conjunto Q para vectores de probabilidad en $\mathbb{R}^3$. Además de tres de los puntos de $\mathcal{N}_1$ (indicados con rombos), se observa sólo un candidato a $\tilde{\theta}_d$ en la región delimitada por las rectas tangentes.

Observación. El hecho de tomar tangentes por los puntos de $\mathcal{N}_{l_0}$ y sólo por ellos, es arbitrario. Podemos elegir planos tangentes por puntos en Θ_k^d cualesquiera, aunque es posible que alguno(s) resulten redundantes. Asimismo no sabemos mucho acerca del tamaño del conjunto Q^d excepto que es menor que $|\Theta_k^d|$ (lo deseable es que sea mucho menor). Podemos agregar entonces un paso extra en nuestra construcción de Q^d: Elegimos un umbral c tal que si $|Q^d| > c$ agrego más planos tangentes a la construcción, es decir

$$Q^d(c) = \left(\bigcap_{x \in \mathcal{N}_{l_c}} Q_x \right) \cap \Theta_k^d,$$

con

$$l_c := \begin{cases} \min\limits_{1 \le l \le d/2} \left\{ l : \# \left(\bigcap\limits_{x \in \mathcal{N}_l} Q_x \right) \cap \Theta_k^d \le c \right\}, & \text{si existe tal valor de } l, \\ \frac{d}{2}, & \text{en otro caso.} \end{cases}$$

Observación. El principal objetivo de la construcción anterior fue reducir el número de puntos "candidatos" a $\tilde{\theta}_d$, debido al gran costo de realizar los cálculos sobre todo Θ_k^d. En consecuencia, es importante diseñar una estrategia eficiente para recorrer $Q^d(c)$ que no involucre verificar si cada punto de Θ_k^d pertenece a este nuevo conjunto. En el apéndice 1 se describe con detalle dicha estrategia.

Una vez hallados los $\tilde{\theta}_d$, podemos optimizar en d el largo de descripción para cada modelo. Para ello calculamos $L_{total}(D, \tilde{\theta}_d, d)$ con $\log(d) \in \mathbb{Z}^+$ hasta que la función cambie de crecimiento, entendiendo que la función es decreciente en d hasta llegar al mínimo y luego crece.[1] Constataremos experimentalmente este comportamiento en el capítulo 4, implementando esta nueva estrategia para distintos modelos y conjuntos de datos.

En resumen, para cada familia paramétrica de modelos M tenemos largo de descripción

$$L_{total}(D, M) = -\log \mathbf{P}_{\tilde{\theta}_{d^*}}(D) + (k+1)\log(d^*),$$

con $\tilde{\theta}_d$ calculado como antes y d^* elegido de modo que

$$d^* = \min_{d:\log(d)\in\mathbb{Z}^+} \{d : L_{total}(D, M, d) < L_{total}(D, M, 2d)\}.$$

Notemos que el largo de descripción total definido en esta sección, si bien es distinto a Crude two-part MDL, presenta un comportamiento similar respecto al tamaño de muestra. Nuevamente, si tenemos un conjunto de datos grande, el término $-\log \mathbf{P}_M(D)$ será también muy grande y se justifica el uso de modelos con mayor cantidad de parámetros. En el siguiente capítulo veremos que en la práctica esto hace que el modelo M_6 prevalezca sobre los otros, incluso M_5 que parece hacer un uso más inteligente de la teoría musical y es el preferido por Temperley, y que los modelos jerárquicos refinados.

[1] Esto se interpreta como que lo que "ahorramos" al tomar un mejor código ya no compensa el costo extra al representar los parámetros.

Capítulo 4

Experimentos y resultados de comparación de modelos

4.1. Consideraciones metodológicas

4.1.1. Construcción del conjunto de datos

Dedicaremos este capítulo a la aplicación de los modelos planteados anteriormente sobre algunos corpus a fin de comparar su ajuste en cada caso. Comenzaremos discutiendo algunos aspectos de la selección y manipulación de las obras a analizar.

Elegiremos los corpus de modo que todas sus piezas tengan la misma estructura métrica (es decir, el mismo compás), ya que varios de los modelos dependen de dicha estructura. Una posible estrategia es, precisamente, considerar aquellas obras que estén completamente en el compás considerado, descartando aquellas que presenten cambios de compás. En la práctica esto resulta a veces un poco restrictivo, ya que se descartan obras que, por ejemplo, presentan cambios de compás muy breves y/o que se escriben para representar pausas o anacrusas sin que realmente haya un cambio estructural. El hecho de procesar los corpus de manera automática hace que sea difícil identificar cuál es el motivo de cada anotación de un cambio de compás.

Un problema similar se presenta con la subdivisión, pues decidimos tomar pulsos de corchea en todos los casos. Los corpus elegidos presentan, ocasionalmente, ataques que no coinciden con ningún pulso (generados, por ejemplo, por figuras con duraciones menores

a la de éste). Nuevamente podríamos considerar la solución más restrictiva: excluir del corpus las obras con ataques fuera de la grilla considerada. Otra alternativa consiste en elegir la unidad de tiempo para los pulsos en función del corpus, lo cual suele derivar en grillas métricas demasiado finas y un cálculo de parámetros excesivamente costoso.[1]

El criterio de excluir por completo todas las obras que incumplan alguna de las condiciones necesarias hace mermar significativamente el tamaño de algunos corpus. Por otra parte, notemos que bajo todos los modelos considerados tiene sentido considerar cada compás como una "unidad" en el siguiente sentido:

- En los modelos M_1 a M_4 los datos pueden verse como una sucesión i.i.d. de compases. En particular, compases iguales tienen la misma probabilidad.
- En los modelos restantes los compases pueden modelarse como una cadena de Markov homogénea de orden 1.

Resulta de utilidad entonces conservar fragmentos de obras en caso de no poder utilizar la obra completa, tomando como unidad mínima el compás. Así en vez de omitir las obras que no cumplan las condiciones necesarias para los modelos, omitimos los compases que no lo hacen. Esta construcción puede llevar a la aparición de compases consecutivos que en realidad no lo son, lo cual no representa un problema en los modelos que asumen compases i.i.d. pero afecta al calculo de probabilidades en los que asumen que los compases se comportan como una cadena de Markov. Utilizaremos de todos modos esta estrategia entendiendo que las falsas transiciones entre compases que se observen no serán atípicas: Más aún, en el caso de los modelos jerárquicos, la única información que se utiliza del compás adyacente es si hay o no un ataque en el primer tiempo (para determinar el tipo de anclaje del segundo pulso más relevante). En particular esto implica que si la probabilidad de ataque en el primer pulso de cada compás es 0 o 1, los compases son también independientes.

Se construye así el conjunto de datos D sobre el cual aplicamos los modelos: para cada obra del corpus elegido se toman los compases que satisfacen las condiciones necesarias para el modelado (es decir, tienen la estructura métrica establecida y no presentan ataques fuera de la subdivisión elegida). Cada uno de ellos se escribe como una secuencia de n_{bpc} unos y ceros, con los unos indicando los lugares de ataque.[2]

[1] Incluso luego de la optimización discutida en la sección 3.2 del capítulo anterior, el costo de optimizar sobre vectores de probabilidad crece rápidamente al incrementar su dimensión.

[2] Recordar que n_{bpc} es la cantidad de pulsos por comás.

Tanto la construcción de D como el posterior cálculo de los largos de descripción correspondientes se realizaron utilizando el lenguaje de programación *python*. En particular usamos la biblioteca *music21* ([Micb]) para procesar los corpus. En [rep] se encuentra disponible el código que permite replicar los experimentos realizados y eventualmente incorporar otros corpus.

4.1.2. Descripción de los corpus

Respecto a la elección de los corpus notemos en primer lugar que tomamos como caso de estudio ritmos de melodías principales, no de acompañamientos o contracantos. Consideramos así obras a una voz, canciones o melodías símil canción. Necesitamos además disponer de dicho repertorio en un formato simbólico adecuado para el análisis automático, el cual no podemos hacer directamente a partir de un audio o una partitura cualquiera. Recordemos además que, incluso ante un corpus con un formato adecuado y de una fuente confiable, se deben filtrar las obras según su estructura métrica y la grilla de pulsos considerada. Tras ese proceso, esperamos tener aún una cantidad de datos razonablemente grande.

Así, utilizamos como fuente parte del corpus incorporado en la propia biblioteca music21 y agregamos un corpus externo de melodías de tangos cantados. Considerando además la distinción según el compás, tenemos un total de 7 corpus distintos que son:

- Las obras en $\frac{4}{4}$ y en $\frac{2}{4}$ de *Essen Folksong Collection* (considerados como dos conjuntos diferentes).
- Las obras en $\frac{4}{4}$ y en $\frac{2}{4}$ del corpus *Aird's Airs* (nuevamente, se divide en dos corpus de acuerdo al compás).
- Las obras en $\frac{4}{4}$ y en $\frac{2}{4}$ del corpus *O'Neills Music of Ireland*.
- Las obras en $\frac{4}{4}$ del *Cancionero del Tango* de Hugo Satorre.

A continuación describiremos con más detalle los corpus seleccionados.

- **Essen Folksong Collection** [Ess]. Se trata de una recopilación de más de 6000 melodías populares mayoritariamente europeas. Se encuentran incorporadas al corpus propio de music21 y contienen obras en distintos compases. De esta colección de obras tomamos dos subconjuntos:

- Fragmentos en $\frac{4}{4}$: Consideramos todos los compases de $\frac{4}{4}$ (sin ataques fuera de la grilla de corcheas). Obtenemos así un total de 23559 compases, tomados de 2104 obras diferentes.
- Fragmentos en $\frac{2}{4}$: De modo similar tomamos los compases de $\frac{2}{4}$, obteniendo 27942 compases de un total de 2533 obras.

- **Aird's Airs** [Air]. Publicada desde 1782 por James Aird , es una selección de aires mayormente de Escocia, Inglaterra e Irlanda. Consiste de 6 volúmenes con un total de 1180 obras (200 en cada uno de los primeros 5 volúmenes, 180 en el sexto). Pese a que entendemos por aires melodías monofónicas, canciones o símil canción, pueden encontrarse 8 dúos en el volumen 6, los cuales excluiremos del análisis. Obtenemos este corpus también del corpus local de music21 y distinguimos según la estructura métrica obteniendo:
  - Fragmentos en $\frac{4}{4}$: Se obtuvieron un total de 2716 compases en $\frac{4}{4}$, provenientes de 307 obras.
  - Fragmentos en $\frac{2}{4}$: En este caso son 2997 de compases, provenientes de 230 obras diferentes.
- **O'Neill's Music of Ireland** [O'N]. Se trata de una colección de 1850 melodías populares irlandesas, recopiladas originalmente en 1903 por Francis O'Neill y disponibles en el corpus local de music21. De éstas luego de filtrar según la estructura métrica y la grilla de pulsaciones nos quedan:
  - Fragmentos en $\frac{4}{4}$: Un total de 2332 compases, tomados de 214 piezas del corpus.
  - Fragmentos en $\frac{2}{4}$: Un total de 933 compases, tomados de 165 piezas.

La disponibilidad de estos corpus en el repositorio local del software utilizado (puede verse una lista detallada de estos y otros corpus en [Mica]) y el hecho de tratarse de melodías populares que se ajustan al tipo de pieza buscado son las principales razones por las que los elegimos para el análisis. Además en el caso del corpus Essen se destaca su gran tamaño, tanto en obras de $\frac{4}{4}$ como en $\frac{2}{4}$. Por otra parte creemos de interés cultural incorporar al estudio un repertorio de música local, lo cual motiva la elección de nuestro último corpus.

- **"Cancionero del Tango" de Hugo Satorre** [Sat]. Se trata de una selección de 204 tangos-canción, presentados como la melodía principal y el cifrado correspondiente.[1] Este corpus, concebido como un insumo para músicos intérpretes, apunta

[1] Dentro de esos "tangos" se encuentran también algunas milongas y valses.

además a ser representativo de la denominada "época de oro" del tango. Dada la especificidad de la elección del repertorio, la calidad de las transcripciones y el hecho de que la mayor parte de las obras estén en $\frac{4}{4}$, pudimos conservar un conjunto de datos razonablemente grande pese a que el corpus original no lo es tanto. Por otra parte, consideramos en este caso sólo obras en $\frac{4}{4}$ pues al tratarse de un corpus fundamentalmente de tangos no hay una cantidad significativa de obras en otros compases. Nos quedan así 6020 compases, provenientes de 173 obras distintas.

En la tabla 4.1 se resume la información numérica de los corpus estudiados.

	Corpus en $\frac{4}{4}$				Corpus en $\frac{2}{4}$		
	Essen	Aird's	O'Neill's	Tangos	Essen	Aird's	O'Neill's
Cantidad de compases	23559	2716	2332	6020	27942	2997	933
Cantidad de obras	2104	307	214	173	2533	230	165

Tabla 4.1: Cantidades de compases y obras considerados en cada corpus.

Veamos a continuación los resultados obtenidos al aplicar los modelos previamente estudiados sobre estos corpus.

4.2. Resultados obtenidos para la comparación de modelos

Hemos descrito hasta ahora dos posibles estrategias para comparar modelos, ambas basadas en MDL: Crude two-part MDL y la versión refinada de éste. Bajo ambos enfoques calculamos el largo de descripción de cada uno de nuestros corpus. Los conjuntos de datos D descritos en los cálculos del capítulo anterior se construyen a partir de las secuencias de ataques de cada uno de ellos. Así, tendremos 7 conjuntos de datos D, secuencias de ceros y unos con tantos números como pulsos haya en total en el corpus. Esto es, las cantidades de compases antes mencionadas multiplicadas por 4 u 8 dependiendo de si es un corpus en $\frac{2}{4}$ o $\frac{4}{4}$ respectivamente.

Los largos de descripción obtenidos dependerán del tamaño del corpus de modo que para normalizar los resultados presentaremos el largo medio de descripción *por compás*, es decir, para cada modelo M presentaremos

$$\overline{L}_{medio}(D, M, d) := \frac{n_{bpc}}{N} L_{total}(D, M, d).$$

De este modo los valores obtenidos para los diferenes corpus (de igual estructura métrica) no deberían diferir tanto, y a su vez se preserva el ordenamiento en la comparación de modelos ya que el factor n_{bpc}/N es común bajo un mismo corpus.

4.2.1. Crude Two-Part MDL

Recordemos que en este caso, el largo medio de descripción por compás resulta

$$L_{medio}(D, M, d) = \frac{n_{bpc}}{N} \left((k+1) \log(d) - \log \mathbf{P}_M(D) \right),$$

con k la cantidad de parámetros del modelo M. Tomando $d = \sqrt{N}$ obtenemos los resultados que se presentan en la tabla 4.2.

Corpus	Corpus en $\frac{4}{4}$				Corpus en $\frac{2}{4}$		
Modelo	Essen	Aird's	O'Neill's	Tangos	Essen	Aird's	O'Neill's
N	188472	21728	18656	48160	111768	11988	3732
$\mathbf{M_1}$	7.999	7.459	7.491	7.567	3.513	2.852	3.316
$\mathbf{M_2}$	7.131	6.463	6.656	6.890	3.424	2.778	3.144
$\mathbf{M_3}$	6.183	6.739	6.733	7.001	3.800	2.509	3.028
$\mathbf{M_4}$	5.943	6.600	6.497	6.825	3.061	2.505	3.031
$\mathbf{M_5}$	5.619	5.508	5.604	6.302	2.901	2.279	2.684
$\mathbf{M_6}$	5.201	4.965	4.975	5.522	2.726	2.138	2.347
$\mathbf{M_7}$	5.479	5.382	5.357	6.005	2.886	2.276	2.673
$\mathbf{M_{total}}$	5.444	5.360	5.351	5.968	-	-	-

Tabla 4.2: Largos medios de descripción (en bits por compás) utilizando Crude Two-Part MDL, con $d = \sqrt{N}$. En el caso de los corpus en $\frac{2}{4}$ el modelo M_7 coincide con el refinamiento total. Para cada corpus se indica también el tamaño de muestra N correspondiente.

Observación. Omitimos mostrar los largos de descripción para otros valores de d en tanto los resultados no presentan mayor interés. Sabemos que el largo de descripción es creciente en d e incluso para valores de d pequeños los modelos se ordenan de la misma forma en cuanto a su desempeño.

Tenemos entonces que para todos los corpus considerados el modelo que más comprime los datos es M_6, seguido de los modelos refinados M_{total} y M_7, para luego ir descendiendo desde M_5 hasta M_1. Pese a que los distintos corpus presentan tamaños bastante distintos, en todos los casos el modelo con menor largo de descripción es el más complejo (M_6). Asimismo el largo medio de descripción por compás no parece reducirse ante el uso de conjuntos de datos más grandes.

Veamos entonces qué ocurre si consideramos una versión reducida de cada corpus, digamos, 100 compases de cada uno. Presentamos los nuevos resultados en la tabla 4.3.

Corpus	Corpus en $\frac{4}{4}$				Corpus en $\frac{2}{4}$		
Modelo	Essen	Aird's	O'Neill's	Tangos	Essen	Aird's	O'Neill's
N	800	800	800	800	400	400	400
$\mathbf{M_1}$	7.339	7.643	6.881	7.453	4.105	2.416	3.155
$\mathbf{M_2}$	5.881	7.187	5.886	6.979	4.092	2.399	3.003
$\mathbf{M_3}$	5.076	6.805	6.522	6.883	3.607	2.331	2.869
$\mathbf{M_4}$	5.138	6.620	6.179	6.793	3.695	2.393	2.834
$\mathbf{M_5}$	5.040	6.116	5.333	6.210	3.694	2.375	2.802
$\mathbf{M_6}$	4.387	5.281	4.315	5.470	3.576	2.327	2.828
$\mathbf{M_7}$	5.539	6.460	5.260	6.599	3.926	2.638	2.895
$\mathbf{M_{total}}$	5.585	6.542	5.399	6.645	-	-	-

Tabla 4.3: Largos medios de descripción (en bits por compás) utilizando Crude Two-Part MDL, con $d = \sqrt{N}$, para los corpus reducidos. En el caso de los corpus en $\frac{2}{4}$ el modelo M_7 coincide con el refinamiento total. Para cada corpus se indica también el tamaño de muestra N correspondiente.

En este caso el ordenamiento de los modelos varía ligeramente: aunque en todos los casos sique siendo M_6 el modelo con mayor compresión, los modelos M_7 y M_{total} resultan ahora menos eficientes que M_5. Es decir que para estos corpus más pequeños el costo de representar los parámetros extra en los modelos jerárquicos refinados no compensa la posible mejora en la representación de los datos. Así, en este caso nuestros modelos efectivamente no tienen el mismo desempeño para todos los corpus.

Veremos a continuación qué ocurre al considerar el truncamiento de los parámetros según d en el cálculo del largo de descripción.

4.2.2. Crude Two-Part MDL refinado con precisión d óptima

En este caso tenemos, para cada conjunto de datos D, el siguiente largo medio de descripción por compás:

$$L_{medio}(D, M, d) = \frac{n_{bpc}}{N} \left(-\log \mathbf{P}_{\tilde{\theta}_d}(D) + (k+1)\log(d) \right),$$

donde nuevamente k es la cantidad de parámetros del modelo y d la precisión al cuantizarlos. Esta vez elegiremos el d óptimo para cada modelo de acuerdo a lo discutido en el capítulo anterior. Nótese que que a diferencia del criterio utilizado en Crude Two-Part MDL, los valores de d óptimos pueden cambiar según el modelo considerado.

En primer lugar fijamos los modelos y los datos y calculamos $L_{medio}(D, M, d)$ en función de d, observando que en cada caso la función tiene el tipo de crecimiento esperado (decreciente y después creciente).[1] En la figura 4.1 se presentan todos los gráficos, a fin de poder comparar los diferentes modelos.

Dado que M_1 y en algunos casos M_2 arrojan valores significativamente superiores a los demás modelos, en la figura 4.2 rehacemos las gráficas omitiéndolos cuando corresponda para obtener una mejor visualización.

Podemos observar cómo efectivamente la elección de d puede afectar el ordenamiento de los modelos, incluso al punto de obtener distintos modelos óptimos. Así, si bien en general suele ser M_6 el modelo que logra mayor compresión, al considerar valores pequeños de d esto cambia. A modo de ejemplo, para $d = 2^5$ todos los corpus en $\frac{4}{4}$ alcanzan la mayor compresión a través de M_{total}.

Para cada caso (M y D fijos) consideraremos d^* minimizando $L_{medio}(D, M, d)$. Luego este largo será el mínimo largo de descripción para D con el modelo en cuestión. Obtenemos así los resultados presentados en la tabla 4.4

4.3. Consideraciones sobre los resultados

Destacaremos a continuación algunos aspectos interesantes que se observan en los resultados de nuestro experimento.

[1] Con excepción de M_1 en algunos corpus. No nos detenemos en la discusión de este modelo en tanto su ajuste es muy malo y su largo de descripción apenas varía con d.

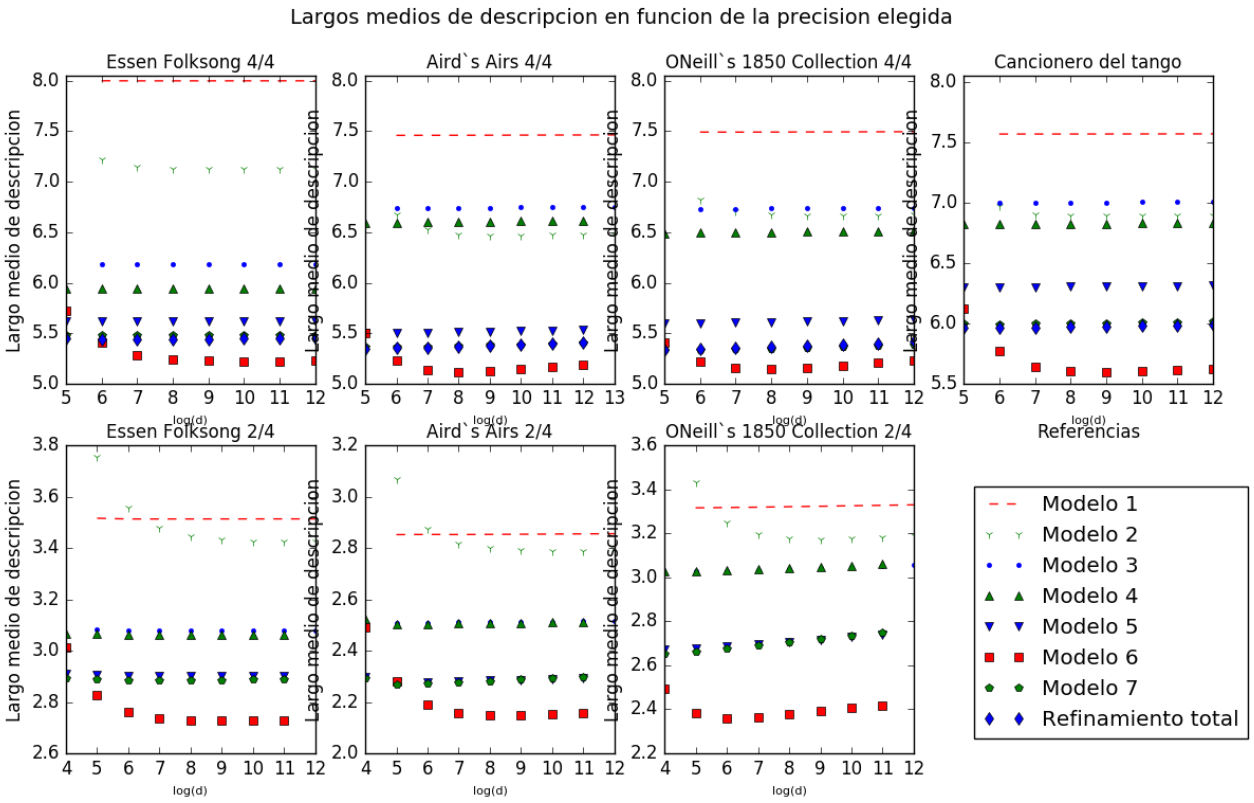

Figura 4.1: Largo medio (en bits por compás) de descripción en función de $\log(d)$ para cada uno de los corpus y modelos estudiados.

- **El modelo ganador.** Para ambos criterios (Crude Two-Part MDL y Crude Two-Part MDL refinado) y para todos los corpus estudiados es M_6 el modelo que minimiza el largo de descripción. Sí ocurre en algunos corpus un cambio en el ordenamiento de los modelos respecto a lo obtenido en Crude Two-Part MDL. En la tabla 4.5 se ordenan los modelos según el largo de descripción que asignan en cada caso. Puede verse que usando Crude Two-Part MDL refinado, el modelo M_2 parece funcionar mejor que M_3 para los corpus en $\frac{4}{4}$, con excepción del Essen Corpus. Este hecho, que en nuestro primer análisis omitimos al tomar precisión d fija, tiene un correlato en lo que ocurre entre los modelos M_6 y M_5 donde el modelo de posición métrica es más eficaz que el que utiliza la estructura de niveles métricos.

  Cabe notar que el modelo M_2 tiene $T + 1$ parámetros siendo T el mayor tiempo inter-ataque observado en el corpus. Sin embargo, por razones prácticas admitimos un máximo de 8 parámetros.[1] En la práctica este truncamiento apenas modifica el vector de parámetros, ya que son poco frecuentes los tiempos inter-ataque tan largos.

[1] Es decir, sólo se consideran los intervalos interataque de hasta 7 pulsos y se normaliza el vector de tasas resultante.

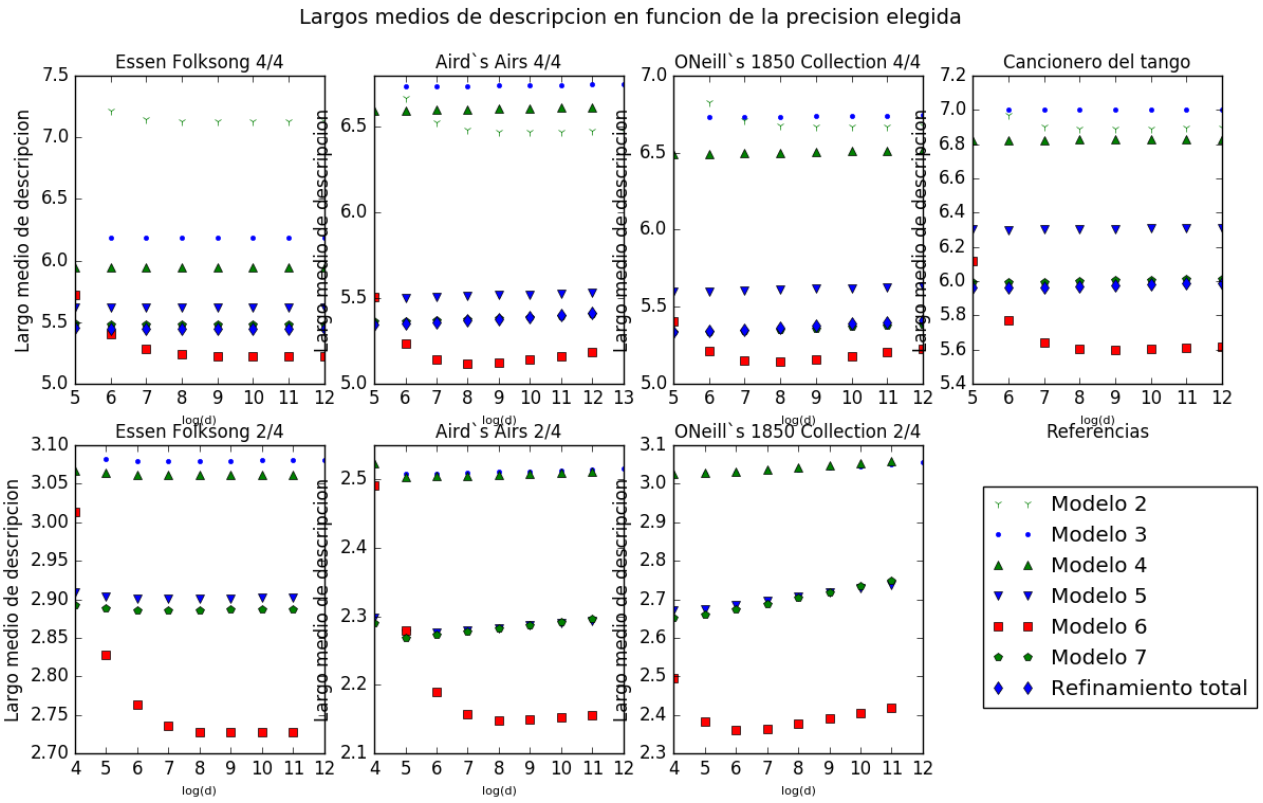

Figura 4.2: Largo medio (por compás) de descripción en función de $\log(d)$ de cada uno de los corpus, para todos los modelos excepto M_1 y M_2. Este último se grafica sólo en los corpus en $\frac{4}{4}$.

Más aún en los corpus en $\frac{2}{4}$ considerados no se observaron tiempos-interataque mayores a 7 por lo que no se realizó truncamiento alguno.

- **Valores de d óptimos.** Como se aprecia en las gráficas de las figuras 4.1 y 4.2, el largo de descripción óptimo se alcanza en distintos valores de d según el modelo y el corpus. Se puede notar de todos modos que hay una relación con el tamaño de muestra, obteniendo en general valores de d que crecen con N. Esto resulta coherente con el criterio tomado en Crude Two-Part, donde definimos $d = \sqrt{N}$. Tambien se observa que los d óptimos en los modelos M_2 y M_6 son particularmente grandes, comparados con los demás modelos sobre el mismo corpus. Recordemos que M_2 y M_6 son los únicos modelos en los que los parámetros forman vectores de probabilidad, con lo que cada $\tilde{\theta}_d$ puede estar bastante lejos de $\hat{\theta}$. Esto hace que tengamos que tomar una precisión d alta para que los datos queden codificados con una buena compresión.
- **Largos de descripción en Crude Two-Part MDL con y sin refinar.** Recordemos que la diferencia entre Crude Two-Part MDL y su versión refinada está en la elección de los parámetros utilizados para codificar los datos. Resulta de este modo

Corpus	Corpus en $\frac{4}{4}$								Corpus en $\frac{2}{4}$					
Modelo	Essen		Aird's		O'Neill's		Tangos		Essen		Aird's		O'Neill's	
$N(\lfloor\sqrt{N}\rfloor)$	188472(434)		21728(147)		18656(136)		48160(219)		111768(334)		11988(109)		3732(61)	
	L_{medio}	d	L_{medio}	d	L_{medio}	d	L_{medio}	d	L_{medio}	d	L_{medio}	d	L_{medio}	d
M_1	7.999	2^7	7.458	2^5	7.491	2^5	7.567	2^6	3.513	2^6	2.851	2^4	3.311	2^3
M_2	7.131	2^9	6.471	2^9	6.667	2^{10}	6.892	2^8	3.425	2^{12}	2.789	2^{10}	3.172	2^9
M_3	6.182	2^7	6.737	2^5	6.731	2^5	7.001	2^6	3.080	2^8	2.508	2^5	3.023	2^4
M_4	5.943	2^7	6.595	2^5	6.493	2^5	6.823	2^6	3.061	2^8	2.504	2^5	3.0274	2^4
M_5	5.618	2^7	5.503	2^5	5.598	2^5	6.298	2^6	2.901	2^6	2.275	2^5	2.671	2^4
M_6	5.223	2^{10}	5.117	2^8	5.145	2^8	5.596	2^9	2.727	2^{10}	2.148	2^8	2.359	2^6
M_7	5.476	2^7	5.364	2^6	5.334	2^5	5.993	2^6	2.885	2^6	2.269	2^5	2.653	2^4
M_{total}	5.442	2^7	5.348	2^5	5.336	2^5	5.960	2^6	-	-	-	-	-	-

Tabla 4.4: Largos medios de descripción (en bits por compás) utilizando Crude Two-Part MDL refinado, se indica en cada caso el argumento óptimo d que da lugar a dicho valor.

Corpus	Corpus en $\frac{4}{4}$								Corpus en $\frac{2}{4}$					
Modelo	Essen		Aird's		O'Neill's		Tangos		Essen		Aird's		O'Neill's	
Método	CTP	CTP-r	CTP	CTP-r	CTP	CTP-r	CTP	CTP-r	CTP	CTP-r	CTP	CTP-r	CTP	CTP-r
Mayor compresión	M_6	M_6	M_6	M_6	M_6	M_6	M_6	M_6	M_6	M_6	M_6	M_6	M_6	M_6
	M_{total}	M_{total}	M_{total}	M_{total}	M_{total}	M_{total}	M_{total}	M_{total}	M_7	M_7	M_7	M_7	M_7	M_7
	M_7	M_7	M_7	M_7	M_7	M_7	M_7	M_7	M_5	M_5	M_5	M_5	M_5	M_5
	M_5	M_5	M_5	M_5	M_5	M_5	M_5	M_5	M_4	M_4	M_4	M_4	M_4	M_4
	M_4	M_4	M_4	M_4	M_4	M_4	M_4	M_4	M_3	M_3	M_3	M_3	M_3	M_3
	M_3	M_3	M_3	M_2	M_3	M_2	M_3	M_2	M_2	M_2	M_2	M_2	M_2	M_2
	M_2	M_2	M_2	M_3	M_2	M_3	M_2	M_3	M_1	M_1	M_1	M_1	M_1	M_1
Menor compresión	M_1	M_1	M_1	M_1	M_1	M_1	M_1	M_1	-	-	-	-	-	-

Tabla 4.5: Ordenamiento de los modelos según su desempeño usando Crude Two-Part MDL (CTP) y Crude Two-Part MDL refinado (CTP-r).

que, para igual modelo, corpus y precisión, la versión refinada de Crude Two-Part MDL debería darnos largos de descripción mayores a la versión no refinada. Los resultados obtenidos no contradicen esto: si bien en algunos casos puede verse como un mismo modelo y corpus obtiene largos de descripción más pequeños con el enfoque refinado, esto puede explicarse por la diferencia en la precisión d. Por otra parte, notamos que si bien el largo de descripción obtenido puede ser menor utilizando una estrategia u otra, en cualquier caso se obtienen valores razonablemente similares en ambos casos, para todos los corpus y modelos. A modo de síntesis presentamos en la tabla 4.6 los largos medios para los modelos más eficientes y las precisiones correspondientes usando Crude Two-Part con y sin refinar.

Tenemos así resutados coherentes al aplicar las dos formas de cálculo de largo de

Modelos	M_4				M_5				M_6				M_{total}			
	L_{medio}		d		L_{medio}		d		L_{medio}		d		L_{medio}		d	
Corpus	CTP	CTP-r	CTP	CTP-r	CTP	CTP-r	CTP	CTP-r	CTP	CTP-r	CTP	CTP-r	CTP	CTP-r	CTP	CTP-r
Essen $\frac{4}{4}$	5.943	5.943	435	2^7 (128)	5.619	5.618	435	2^7 (128)	5.201	5.223	435	2^{10} (1024)	5.444	5.442	435	2^7 (128)
Aird's $\frac{4}{4}$	6.600	6.595	148	2^5 (32)	5.508	5.503	148	2^5 (32)	4.965	5.117	148	2^8 (256)	5.360	5.348	148	2^5 (32)
O'Neill's $\frac{4}{4}$	6.497	6.493	137	2^5 (32)	5.604	5.598	137	2^5 (32)	4.975	5.145	137	2^8 (256)	5.351	5.336	137	2^5 (32)
Tangos	6.825	6.823	220	2^6 (64)	6.302	6.298	220	2^6 (64)	5.522	5.596	220	2^9 (512)	5.968	5.960	220	2^6 (64)
Essen $\frac{2}{4}$	3.061	3.061	335	2^8 (256)	2.901	2.901	335	2^6 (64)	2.726	2.727	335	2^{10} (1024)	2.886	2.885	335	2^6 (64)
Aird's $\frac{2}{4}$	2.505	2.504	110	2^5 (32)	2.279	2.275	110	2^5 (32)	2.138	2.148	110	2^8 (256)	2.276	2.269	110	2^5 (32)
O'Neill's $\frac{2}{4}$	3.031	3.0274	62	2^4 (16)	2.684	2.671	62	2^4 (16)	2.347	2.359	62	2^6 (64)	2.673	2.653	62	2^4 (16)

Tabla 4.6: Largo medio (en bits por compás) para los modelos M_4,M_5, M_6 y M_{total} (en el caso de $\frac{2}{4}$ sería M_7), usando Crude Two-Part MDL (CTP) y Crude Two-Part MDL refinado (CTP-r)

descripción consideradas que además nos llevaron a la misma conclusión respecto a cuál es el modelo que mejor se ajusta a cada uno de nuestros corpus (aunque dicha conclusión no tiene que ser necesariamente la misma). En cuanto a los modelos, nuestros criterios eligen M_6 por sobre los demás pese a la gran cantidad de parámetros que éste tiene, incluso cuando reducimos significativamente el tamaño del corpus como forma de penalizar el uso de modelos muy complejos.

Capítulo 5

Conclusiones

Sobre los resultados obtenidos

En esta tesis abordamos el problema de simulación de melodías, modelando tanto las alturas de las notas, como su dimensión rítmica. Para ello, retomamos un trabajo previo sobre simulación de alturas sobre lo cual incorporamos modelos para simular ritmos. Luego, a modo de ejemplo, utilizamos dichos modelos para simular melodías aleatorias basadas en una melodía preexistente (Arroz con Leche), cuya armonía conocemos.

Las melodías obtenidas resultan, desde una evaluación subjetiva, rítmica y tonalmente coherentes, posiblemente debido al restrictivo criterio utilizado para elegir las alturas. Para las simulaciones de ritmo en las variaciones de Arroz con Leche utilizamos dos modelos distintos (M_5 y M_7) y en ambos casos estimamos los parámetros a partir de las obras en $\frac{2}{4}$ del corpus Essen. Por lo tanto es razonable que ciertos patrones rítmicos que aparecen recurrentemente en la melodía original no aparezcan con tanta frecuencia en las variaciones mientras que, por el contrario, se observan algunos patrones que no se observan en el verdadero Arroz con Leche. Desde el punto de vista rítmico sólo preservamos de nuestra melodía original el compás de $\frac{2}{4}$ y la presencia de ataque en los lugares donde hay restricciones. Todo lo demás se obtiene de un corpus que a priori poco tiene que ver con Arroz con Leche.

Para la simulación de ritmos consideramos distintos modelos posibles. Si bien utilizamos solamente dos de ellos para generar los ejemplos, estudiamos un total de ocho modelos para ritmo. Tal variedad motiva la búsqueda de criterios para comparar modelos, a fin de poder seleccionar el más adecuado (entendido como el que más se adecua a un corpus de

referencia).

El estudio de dichos criterios constituye la parte central de esta tesis. Si bien estas estrategias no nos permiten dar una evaluación de los distintos modelos en términos absolutos, sí nos permite comparar sus ajustes a ciertos conjuntos de datos. Así podemos obtener un posible criterio de selección entre diversos modelos más allá de un criterio exclusivamente estético. Permite además realizar dicha selección sin necesariamente generar ejemplos bajo cada modelo (en nuestro caso de estudio, no fue necesario implementar todos los modelos estudiados para compararlos).

Estudiamos dos criterios para comparar modelos, basados en MDL. En particular el criterio que denominamos Crude Two-Part MDL refinado presenta un enfoque diferente a las variantes de MDL presentes en la bibliografía revisada. Nuestra idea es similar a la propuesta por Barron, Rissanen y Yu en [BRY], donde también se considera la incidencia de la precisión d en el cálculo del largo de descripción del conjunto de datos, pero no se realiza dicho cálculo. Recordemos que el cálculo explícito del largo de descripción es un problema de optimización sobre un dominio discreto posiblemente muy grande, con lo que resulta clave el uso de una estrategia eficiente para hallar el mínimo buscado. Diseñamos y aplicamos un algoritmo que permitió calcular los mínimos largos de descripción, el cual describimos con detalle en el apéndice.

Con respecto a los resultados obtenidos en la comparación de modelos cabe realizar varios comentarios. En primer lugar, nótese que las dos variantes de MDL que empleamos asignaron, para todos los corpus considerados, el mismo modelo óptimo (M_6). Esto no tiene por que ser necesariamente así en general, pudiendo ocurrir que el modelo que ajusta mejor un cierto corpus no sea el mejor para otro. Observamos también que los parámetros estimados para un mismo modelo pueden diferir significativamente entre un corpus y otro. Esto ocurre en nuestros ejemplos de estudio, como puede observarse en la tabla 5.1 donde presentamos a modo ilustrativo los parámetros estimados para el modelo M_5 en los corpus en $\frac{4}{4}$.

Parámetros	$\hat{\theta}_{IV}$	$\hat{\theta}_{III_{00}}$	$\hat{\theta}_{III_{10}}$	$\hat{\theta}_{III_{01}}$	$\hat{\theta}_{III_{11}}$	$\hat{\theta}_{II_{00}}$	$\hat{\theta}_{II_{10}}$	$\hat{\theta}_{II_{01}}$	$\hat{\theta}_{II_{11}}$	$\hat{\theta}_{I_{00}}$	$\hat{\theta}_{I_{10}}$	$\hat{\theta}_{I_{01}}$	$\hat{\theta}_{I_{11}}$
Essen	0.924	0.963	0.882	0.194	0.814	0.036	0.365	0.823	0.693	0.024	0.038	0.367	0.284
Aird's $\frac{4}{4}$	0.930	0	0.982	0.036	0.897	0.005	0.338	0.279	0.823	0.032	0.090	0.463	0.641
O'Neill's $\frac{4}{4}$	0.949	0	0.966	0.017	0.885	0	0.170	0.368	0.781	0.014	0.141	0.428	0.655
Tangos	0.951	1	0.618	0.756	0.771	0.047	0.457	0.483	0.637	0.076	0.258	0.567	0.729

Tabla 5.1: Parámetros del modelo M_5 estimados para los corpus en $\frac{4}{4}$.

En efecto, algunos parámetros difieren significativamente entre corpus, lo que indica que el modelo está detectando diferencias rítmicas entre ellos. Es especialmente notoria la diferencia para el parámetro $\hat{\theta}_{III_{00}}$, que es exactamente 0 en dos de los corpus y 1 o cercano a 1 en los otros dos.[1] Cabe aclarar, sin embargo, que la estimación de dicho parámetro se realiza sobre un total de casos muy pequeño (es decir, en todos los corpus hay muy pocos pulsos de nivel 3 no-anclados sobre los cuales contar ataques). Por otro lado, tenemos parámetros como $\hat{\theta}_{III_{01}}$ y $\hat{\theta}_{I_{11}}$ que además de diferir notoriamente entre corpus fueron estimados sobre una cantidad estadísticamente más significativa de pulsos.

Otro aspecto a destacar respecto a la comparación de modelos es que en todos los casos se prefirió el modelo con más parámetros (M_6), incluso cuando puede no ser el más elegante desde el punto de vista de la teoría musical. De hecho Temperley prefiere el modelo M_5 antes que éste, argumentando que la cantidad de parámetros de M_6 es injustificadamente excesiva [Tem10]. De acuerdo a nuestros experimentos esto no es así, e incluso para versiones reducidas de los corpus sigue siendo conveniente asumir el costo extra de representar los parámetros adicionales.

Pese a esto, el modelo M_6 parece excesivo en cuanto a su complejidad, al menos en el caso de los compases de $\frac{4}{4}$ donde se estima un total de 56 parámetros muchos de los cuales resultan, en la práctica, nulos. Esta situación se da especialmente cuando los corpus no son muy grandes, pero persiste incluso en los más grandes como el corpus Essen. A continuación se presentan las matrices de transición estimadas para dicho corpus y Cancionero del Tango, que son los más grandes dentro de los considerados (en $\frac{4}{4}$):

Essen Folksong Collection

$$\begin{pmatrix}
0.020 & 0.127 & 0.454 & 0.163 & 0.136 & 0.0132 & 0.0843 & 0.0016 \\
0.00178 & \mathbf{0} & 0.960 & 0.0328 & 0.00427 & 0.000712 & 0.000712 & 0 \\
0.00683 & 0 & 0.000706 & 0.297 & 0.611 & 0.0229 & 0.0578 & 0.00314 \\
0.00592 & 0.00161 & \mathbf{0} & \mathbf{0} & 0.982 & 0.00645 & 0.00108 & 0.00255 \\
0.138 & 0.0001648 & 0.00588 & 0 & 0.00472 & 0.157 & 0.611 & 0.0833 \\
0.0392 & \mathbf{0} & 0.000280 & \mathbf{0} & \mathbf{0} & 0.000280 & 0.946 & 0.0140 \\
0.596 & 0.000169 & 0.00186 & 0.000169 & 0.00242 & 0.00197 & 0.0224 & 0.375 \\
0.957 & 0.00192 & 0.0006 & 0.00012 & 0.00168 & 0.0048 & 0.0314 & 0.00264
\end{pmatrix}$$

[1] Recordemos que esta notación fue introducida en el capítulo 2. El parámetro $\theta_{III_{00}}$ es la probabilidad de ataque sobre un pulso de nivel 3 no-anclado.

Notemos que pese a que en este corpus estimamos sobre una gran cantidad de transiciones (188472), tuvimos 9 entradas nulas en la matriz. Además se tienen otras 10 entradas que no son cero pero tienen un valor menor a 10^{-3}. Esta situación es aún más notoria al considerar un corpus más pequeño, como observamos en el siguiente caso.

Cancionero del Tango

$$\begin{pmatrix} 0.0569 & 0.420 & 0.266 & 0.0838 & 0.113 & 0.0293 & 0.0183 & 0.0127 \\ 0.000799 & \mathbf{0} & 0.624 & 0.359 & 0.00999 & 0.00480 & 0.0012 & 0.0004 \\ 0.053 & 0.0061 & 0.000964 & 0.631 & 0.181 & 0.0928 & 0.0254 & 0.00996 \\ 0 & 0 & \mathbf{0} & \mathbf{0} & 0.998 & 0.00178 & 0 & 0 \\ 0.135 & 0.00652 & 0.00391 & 0.00239 & 0 & 0.552 & 0.228 & 0.0724 \\ 0.000659 & \mathbf{0} & 0 & \mathbf{0} & \mathbf{0} & 0 & 0.886 & 0.113 \\ 0.0812 & 0.00964 & 0 & 0.000507 & 0.000254 & 0.000507 & 0.000254 & 0.908 \\ 0.982 & 0.00206 & 0.00183 & 0.00252 & 0.00114 & 0.00389 & 0.00435 & 0.00183 \end{pmatrix}$$

Aquí tenemos 14 entradas nulas, y 7 valores positivos menores a 10^{-3}. Más aún, se observa que hay 6 entradas (indicadas en negrita) en las que ambas matrices se anulan. Estamos considerando un modelo que utiliza muchos parámetros (lo cual en términos del largo de descripción implica costo extra), varios de los cuales no aportan información significativa.

Posibles extensiones

De los resultados de nuestro trabajo se desprenden varias preguntas y posibles líneas de trabajo. Mencionaremos a continuación algunos aspectos sobre los cuales es posible ampliar y/o mejorar lo hecho.

En primer lugar, nuestra simulación de melodías se realizó de un modo bastante sencillo y con múltiples mejoras posibles. Sobre esto queremos destacar que si bien consideramos la altura y el ritmo de forma independiente, es razonable suponer que en realidad no lo son y queda pendiente el planteo de posibles modelos conjuntos para altura y ritmo, por ejemplo considerando la fuerza métrica a la hora de asignar las probabilidades de las diversas alturas.

Por otra parte, para la simulación de ejemplos consideramos como melodía de referencia Arroz con Leche, pero lo realizado puede extenderse sin mayor dificultad a la simulación de variaciones de otras obras de las que se disponga de la armonía anotada utilizando, por ejemplo, un paseo al azar con restricciones sobre los arpegios correspondientes en cada caso. También es posible utilizar escalas asociadas a los acordes en cuestión si se busca un enfoque menos restrictivo. Cabe observar que en la mayoría de los corpus utilizados (así como otros que fueron revisados y no utilizados por diversas razones) no se dispuso de la armonía anotada y la detección automática de ésta constituye un posible tema de estudio en si mismo. Por esta razón elegimos no profundizar en la generalización de estrategias para simular variaciones de melodías dadas a partir de la armonía y unas pocas notas fijas. Destacamos de todos modos que el corpus "Cancionero del Tango" sí presenta la armonía anotada y si bien no la utilizamos para esta tesis, puede tenerse en cuenta para implementar la generalización recién propuesta o incorporarla al análisis de otros modos.[1] Por otra parte, en [Rum17] puede encontrarse ejemplos de simulación de música a partir de corales de J.S. Bach, en los que se admiten todas las notas de la escala cromática y las probabilidades de ocurrencia de cada una se aprenden de los propios corales.

Al respecto de la comparación de modelos, tenemos que ambas estrategias propuestas prefirieron, para todos los corpus estudiados, el modelo con más parámetros. Pese a esto consideramos, en vista de lo antes expuesto, que es posible obtener un modelo menos complejo que M_6 con mejores resultados. Asimismo, los modelos jerárquicos parecen teóricamente más adecuados. La penalización por cada parámetro utilizado en nuestros modelos puede ser excesivamente pequeña y dejamos abierta la discusión de posibles ajustes a realizar sobre ésta. A modo de referencia notemos que en Lasso se tiene un problema de optimización similar, en el cual para prevenir el sobreajuste se incorpora una penalización proporcional a la norma del argumento óptimo que se ajusta según los datos [BvdG11].

En síntesis, presentamos una estrategia de simulación de ritmos, alturas y/o ambas componentes que pese a sus limitaciones arroja resultados razonables. Proponemos también una estrategia de comparación de modelos cuya implementación detallamos de modo que puede ser aplicada a otros modelos. De la implementación realizada sobre distintos corpus destacamos la incorporación al estudio de un corpus de caracter local que puede ser un insumo de interés para futuros trabajos en el área.

[1] El Cancionero del Tango fue publicado durante la fase final de nuestros experimentos, con lo cual nos limitamos a darle el mismo tratamiento que a los demás corpus estudiados, en los que no se disponía de la armonía anotada.

Referencias bibliográficas

[Air] *A selection of Scotch, English, Irish and foreign airs*, http://trillian.mit.edu/~jc/music/book/oneills/1850/, Visitada el 5/4/2019.

[Bar87] Clarence Barlow, *Two Essays on Theory*, Computer Music Journal **11** (1987), no. 4, 56–60.

[BRY] Andrew R Barron, Jorma Rissanen, and Bin Yu, *The Minimum Description Length Principle in Coding and Modeling*, IEEE Trans. Inf. Theory, no. 6, 2743–2760.

[BS03] B. Benward and M.N. Saker, *Music in Theory and Practice*, Music in Theory and Practice, McGraw-Hill, 2003.

[BV04] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge University Press, New York, NY, USA, 2004.

[BvdG11] Peter Bhlmann and Sara van de Geer, *Statistics for High-Dimensional Data: Methods, Theory and Applications*, 1st ed., Springer Publishing Company, Incorporated, 2011.

[Cop] David Cope, *Emily Howell*, http://artsites.ucsc.edu/faculty/cope/Emily-howell.htm, Visitada el 5/4/2019.

[Cop96] ———, *Experiments in Musical Intelligence*, A-R Editions, Inc, 1996.

[CT06] Thomas M. Cover and Joy A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*, Wiley-Interscience, New York, NY, USA, 2006.

[Dic06] Peter. Dickinson, *Cagetalk : dialogues with and about John Cage / edited by Peter Dickinson*, University of Rochester Press Rochester, NY, 2006.

[Ess] *"The Essen Folksong Collection"*, http://essen.themefinder.org/, Visitada el 5/4/2019.

[Gru07] Peter D. Grunwald, *The Minimum Description Length Principle (Adaptive Computation and Machine Learning)*, The MIT Press, 2007.

[JKW17] Daniel Johnson, Robert Keller, and Nicholas Weintraut, *Learning to Create Jazz Melodies Using a Product of Experts*, 2017.

[Joh] Daniel Johnson, *Composing Music With Recurrent Neural Networks*, http://www.hexahedria.com/2015/08/03/composing-music-with-recurrent-neural-networks/, Visitada el 30/1/2017.

[LJ83] Fred Lerdahl and Ray Jackendoff, *A generative theory of tonal music*, The MIT Press, Cambridge. MA, 1983.

[Lun] Lunaverus, *Automatic Music Transcription Software*, https://www.lunaverus.com/home, Visitada el 5/4/2019.

[Mica] *List of Works Found in the music21 Corpus*, http://web.mit.edu/music21/doc/about/referenceCorpus.html, Visitada el 5/4/2019.

[Micb] *music21: A toolkit for computer-aided musicology*, http://web.mit.edu/music21/, Visitada el 5/4/2019.

[NRJB15] Leonardo O. Nunes, Martín Rocamora, Luis Jure, and Luiz W. P. Biscainho, *Beat and Downbeat Tracking Based on Rhythmic Patterns Applied to the Uruguayan Candombe Drumming*, Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Málaga, Spain, October 26-30, 2015, 2015, pp. 264–270.

[O'N] *O'Neill's Music of Ireland*, http://trillian.mit.edu/~jc/music/book/oneills/1850/, Visitada el 5/4/2019.

[Pac] François Pachet, *Flow Machines*, https://www.flow-machines.com/, Visitada el 5/4/2019.

[PRB11] François Pachet, Pierre Roy, and Gabriele Barbieri, *Finite-Length Markov Processes with Constraints*, Proceedings of the Twenty-Second International Joint Conference on Artificial Itelligence (2011), 638–642.

[rep] *Repositorio online*, www.cmat.edu.uy/~vrumbo/repositoriotesis, Visitada el 5/4/2019.

[Ris78] Jorma Rissanen, *Modeling by shortest data description*, Automatica **14** (1978), 465–471.

[Ris83] ———, *A Universal Prior for Integers and Estimation by Minimum Description Length*, The Annals of Statistics **11** (1983), no. 2, 416–431.

[Ris86] ———, *Stochastic Complexity and Modeling*, The Annals of Statistics **14** (1986), no. 3, 1080–1100.

[Rum17] Verónica Rumbo, *Cadenas de Markov con restricciones y aplicación a la composición automática de música tonal*, http://www.cmat.edu.uy/ vrumbo/monografia, 2017, Trabajo monográfico.

[Sat] *Cancionero del tango*, https://drive.google.com/drive/u/0/folders/1kxOXWerFpN-IrQRV7BKVorWxIpvlDq3T?fbclid=IwAR3hG1OydfWRv3h6n1Gc7Bm_s6xmGMOAtIRHxc8NZP1GOMRhkul-aN7vvq0, Visitada el 5/4/2019.

[Sci] *Scientific Pitch Notation*, https://en.wikipedia.org/wiki/Scientific_pitch_notation, Visitada el 5/4/2019.

[Tem07] David Temperley, *Music and Probability*, The MIT Press, 2007.

[Tem10] David Temperley, *Modeling Common-Practice Rhythm*, Music Perception - MUSIC PERCEPT **27** (2010), 355–376.

[Wur] *Würfel-Menuet*, https://archive.org/details/imslp-kirnberger-johann-philipp, Visitada el 5/4/2019.

APÉNDICES

Apéndice 1

Construcción de Q^d

Este apéndice detalla la construcción del conjunto Q^d definido en los cálculos del largo de descripción en MDL refinado (presentado en el capítulo 3). Recordemos que nuestro problema general consiste en minimizar una función convexa que si bien está definida en un conjunto $\Theta_k \subset [0,1]^k$, la restringimos a una grilla finita Θ_k^d de puntos. Contamos para ello con $\hat{\theta}$, mínimo de nuestra función en Θ_k, y con una estrategia para delimitar la ubicación del mínimo $\tilde{\theta}_d$ buscado. Consiste en definir un conjunto Q^d de candidatos a mínimo a través de los hiperplanos tangentes a los conjuntos de nivel de algunos puntos, aprovechando la convexidad de la función.

Describimos el procedimiento empleado para determinar el conjunto Q^d sin visitar todos los puntos de Θ_k^d (lo cual resultaría demsiado costoso incluso para valores de k relativamente pequeños). Utilizamos la notación definida en el capítulo 3, y dado $x \in \Theta_k$ utilizaremos el término *vecinos* para referirnos a puntos de la grilla cuyas coordenadas no estén a más de un paso de distancia, es decir

$$x \in \Theta_k, y \in \Theta_k^d \text{ son vecinos si } ||x-y||_\infty \leq \frac{1}{d}. \tag{1.1}$$

Podemos extender esta noción a la de *vecinos de nivel* r, siendo r un entero positivo, como los puntos que están a r pasos de distancia.[1]

$$x \in \Theta_k, y \in \Theta_k^d \text{ son vecinos de nivel } r \text{ si } \lfloor ||x-y||_\infty \rfloor = \frac{r}{d}. \tag{1.2}$$

[1]Notar que son "exactamente" r pasos de distancia

Consideremos el siguiente esquema inicial de construcción:

- Consideramos un conjunto fijo de puntos, cuyos hiperplanos tangentes se utilizan para delimitar Q^d. Elegimos para ello los vértices del menor hipercubo de puntos en Θ_k^d que contiene a $\hat{\theta}$ (que asumimos no pertenece a Θ_k^d). En la figura 1.1 se ilustra dicha elección para un caso de ejemplo.

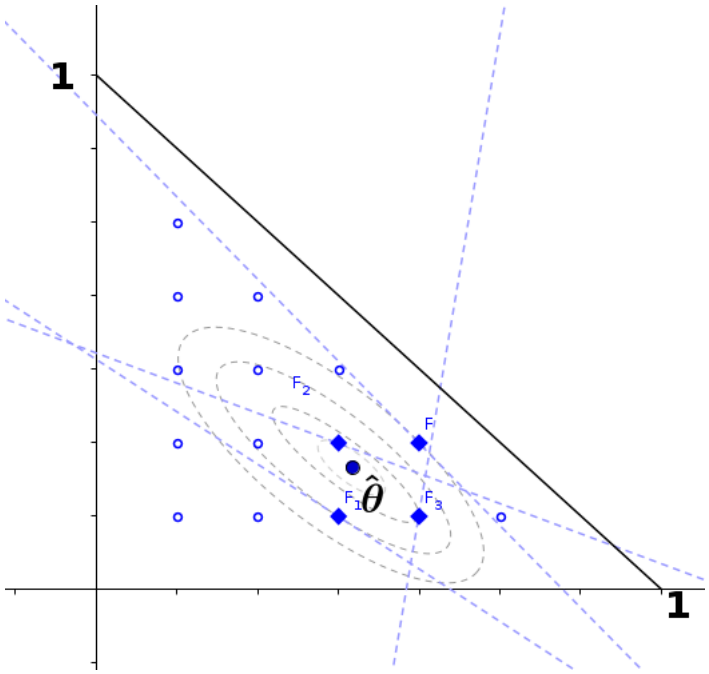

Figura 1.1: Elección de los 2^2 puntos vecinos de $\hat{\theta}$, con sus conjuntos de nivel y rectas tangentes correspondientes, para un caso donde Θ_k son vectores de probabilidad en $\mathbb{R}^3$. Se representan sin relleno todos los puntos de Θ_k^d que aún no han sido visitados.

- Verificamos si los puntos del hipercubo anterior pertenecen a la región delimitada por los hiperplanos seleccionados. Agregamos dichos puntos a Q^d. Definimos también conjuntos auxiliares V y W que iremos modificando, e inicializamos V como los puntos en cuestión (ver figura 1.2).
- El conjunto W es el de los *vecinos a recorrer*, esto es, puntos vecinos a los de V, que no han sido visitados aún y que también pertenecen a Q^d. Así, para cada punto en V visitamos sus vecinos y los agregamos a W si verifican las siguientes condiciones:
  - Pertenecen a la región delimitada por los planos tangentes.
  - No fueron agregados (aún) a Q^d, ni a V.

  Al finalizar la recorrida de V se tiene un conjunto W, posiblemente vacío, de puntos "nuevos" a agregar a Q^d y cuyos vecinos debemos también visitar. Así, agregamos a Q^d los puntos de W y redefinimos $V = W$. En la figura 1.3 se agrega en este paso un conjunto formado por un sólo punto.

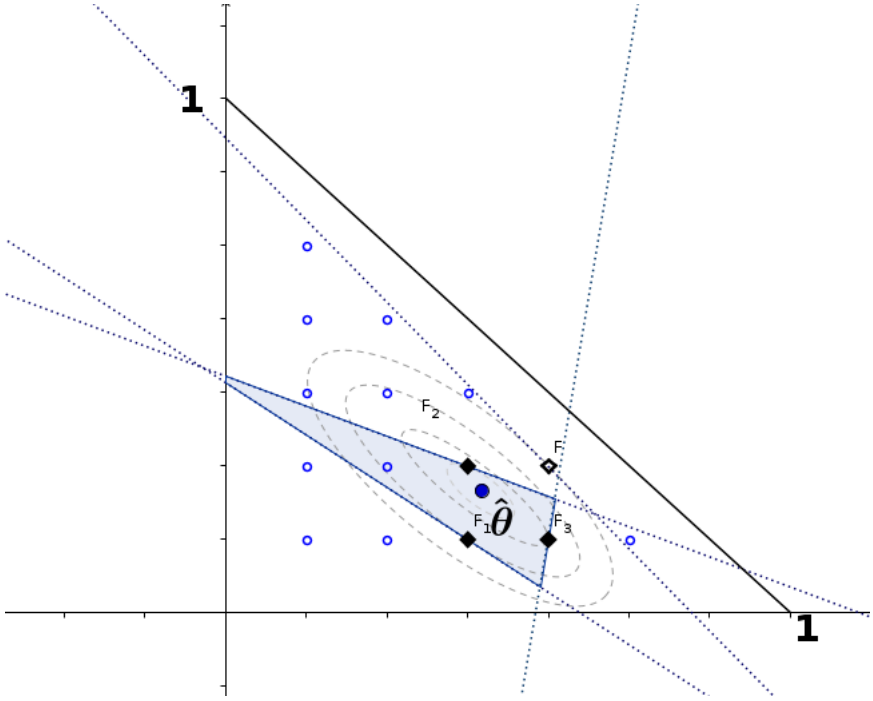

Figura 1.2: Construcción del la región delimitada por las rectas tangentes (sombreado) y primer conjunto V (diamantes con relleno). Se distinguen tras ser examinados los puntos que no son incorporados a Q^d (diamantes sin relleno).

- Visitamos los vecinos de los puntos de nuestro nuevo conjunto V como en el paso anterior, construyendo así un nuevo conjunto W.

El procedimiento termina cuando W es vacío, ya que los puntos que no fueron visitados hasta entonces no pertenecen a Q^d. Sin embargo notemos que la cantidad de vecinos de un punto crece exponencialmente con la dimensión del dominio. Así, es posible que aún sea muy costoso visitar todos los vecinos de los puntos en V. Para acotar la cantidad de puntos a visitar modificamos ligeramente el algoritmo anterior: si en algún paso del algoritmo la cantidad de elementos de W supera cierto umbral preestablecido, incorporamos nuevas rectas tangentes hasta reducir lo suficiente el conjunto W.

Con esta modificación queda establecido el algoritmo para estimar un vector de probabilidad en $\mathbb{R}^k$ por máxima verosimilitud, sujeto a una grilla de d (potencia de 2) valores por coordenada. Presentamos a continuación un algoritmo para calcular el largo medio de descripción por símbolo, según $\hat{\theta}$, d y el umbral U a utilizar, es decir

$$\min_{\theta \in \Theta_k^d} \frac{1}{N} \mathbf{P}_\theta(D).$$

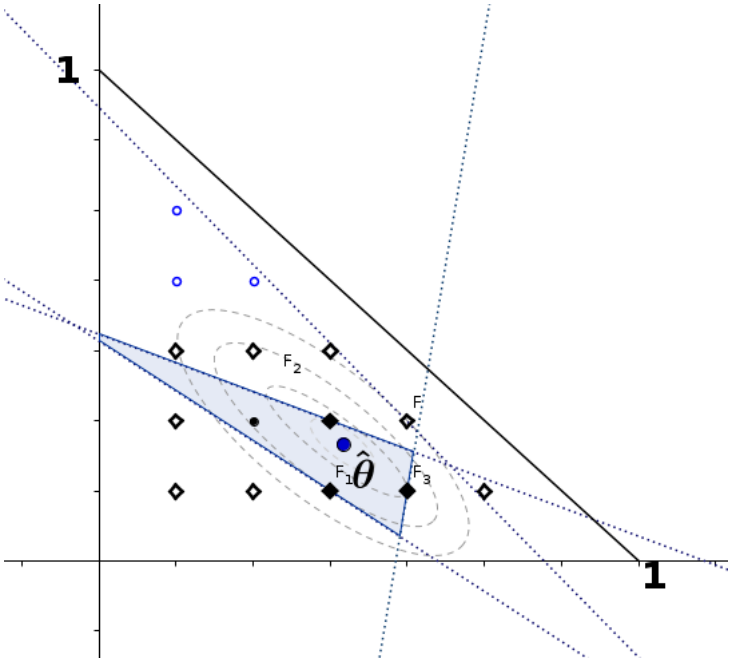

Figura 1.3: Primer conjunto W de vecinos a recorrer. En este caso, será el único W no vacío y el procedimiento termina en el siguiente paso. Los puntos representados como círculos sin relleno son los que nos evitamos visitar con este procedimiento. Finalmente Q^d es el conjunto de los puntos de la grilla con relleno.

A fin de simplificar la lectura consideramos $k \geq 3$ (los casos $k = 1$ y $k = 2$ se resuelven de forma directa) y asumimos dada la función para hallar los vecinos, cuya implementación no reviste mayor complejidad y puede verse directamente en el código. Así, llamamos $\mathcal{N}$ a la función que dado un punto en Θ_k una precisión d y un radio r devuelve el conjunto de sus vecinos de nivel r en Θ_k^d definidos como en la ecuación (1.2), que tienen todas sus coordenadas no nulas.

Procedimiento

Entrada: $\hat{\theta} \in [0,1]^k$ vector de probabilidad con entradas positivas, $d \in \mathbb{Z}^+$ precisión, $U \in \mathbb{Z}^+$ umbral.

- Inicializamos $\mathcal{G} = \left\{(x, v_x) : x \in \mathcal{N}(\hat{\theta}, d, 1),\ v_{xj} = \frac{\theta_k}{1-\sum_{i=1}^{k-1} x_i} - \frac{\theta_j}{x_j}\right\}$, conjunto de los puntos vecinos de $\hat{\theta}$ con sus correspondientes vectores gradientes.
- Establecemos un radio inicial $r = 1$
- Inicializamos $Q^d = \emptyset$ y buscamos los primeros vecinos de $\hat{\theta}$ que pertenezcan a la región delimitada por los hiperplanos tangentes.
- Mientras $Q^d = \emptyset$:
  - Defino $Q^d = \left\{\theta \in \mathcal{N}(\hat{\theta}, d, r) : \langle \theta - x, v_x \rangle < 0, \forall (x, v_x) \in \mathcal{G}\right\}$.

- Actualizo $r = r + 1$.

Al finalizar hemos identificado todos los vecinos de radio menor o igual a cierto r que son candidatos a óptimo. Recorremos ahora los vecinos de dichos puntos.

- Inicializo $V = Q^d$.
- Mientras $V \neq \emptyset$:
  - Inicializo $W = \emptyset$.
  - Para cada $p \in V$:
    - Para cada $\theta \in \mathcal{N}(p, d, 1)$, agrego θ a W si verifica las condiciones:
      - $\theta \notin W$, $\theta \notin V$,
      - $\langle \theta - x, v_x \rangle < 0, \forall (x, v) \in \mathcal{G}$,
  - En caso de que el conjunto W sea demasiado grande, consideramos un criterio más restrictivo. Así, mientras $|W| > U$ y $|\mathcal{G}| < |Q^d|$:
    - Tomo $q \in V$ de modo que $(q, v_q) \notin \mathcal{G}$. Agrego (q, v_q) a $\mathcal{G}$.
    - Para cada $\theta \in W$, si $\langle \theta - q, v_q \rangle < 0$, elimino θ de W.
  - Actualizo $Q^d = Q^d \cup W$ y $V = W$

Salida: El mínimo largo medio de descripción por simbolo dentro del conjunto Q^d hallado, es decir

$$\min_{\theta \in Q^d} \left\{ -\langle \hat{\theta}, \log(\theta) \rangle \right\}.$$

Apéndice 2

Contenido del repositorio

Además de la presente tesis, se encuentra disponible un repositorio online que puede descargarse desde el siguiente link:

http://www.cmat.edu.uy/~vrumbo/repositoriotesis

Detallamos a continuación el material contenido en el repositorio.

- Archivos MIDI de los ejemplos de melodías aleatorias presentadas.
- Código utilizado para simular las melodiás y calcular los distintos largos de descripción.
- Valores calculados para los cálculos de los mínimos largos de descripión, esto es:
  - Estimadores de los parámetros de los distintos modelos para ritmo para los corpus estudiados.
  - Largos de descripción calculados para los distintos valores de d.
- Más ejemplos de variaciones sobre Arroz con Leche.

Software utilizado

Para la simulación de melodías y los cálculos de largos de descripción se utilizaron los siguientes lenguajes y programas, necesarios para ejecutar los scripts provistos en el repositorio.

Se utiliza el lenguaje *python*, con la biblioteca *music21*. Además, se recomienda el software *lilypond* para generar las partituras.

Para la simulación de melodías utilizamos, además de los programas antes mencionados, el lenguaje *R*.