



UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS DE POSTGRADO
COORDINACIÓN DE POSTGRADO EN CIENCIAS DE LA COMPUTACIÓN
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

**MÚSICA INTERACTIVA BASADA EN
IMITACIÓN AUTOMÁTICA DE ESTILO**

Trabajo de Grado presentado a la Universidad Simón Bolívar por
Carlos Esteban Gómez Chacón

Como requisito parcial para optar al grado académico de
Magister en Ciencias de la Computación

Con la asesoría de la Prof.^a
Ivette Martínez

Mayo 2014

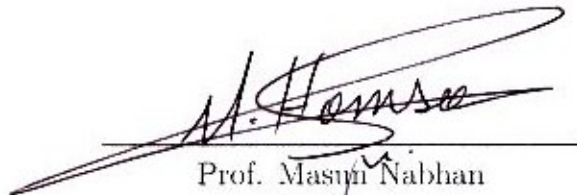


UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS DE POSTGRADO
COORDINACIÓN DE POSTGRADO EN CIENCIAS DE LA COMPUTACIÓN
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

MÚSICA INTERACTIVA BASADA EN
IMITACIÓN AUTOMÁTICA DE ESTILO

Por: Carlos Esteban Gómez Chacón
Carné No.: 07-86055

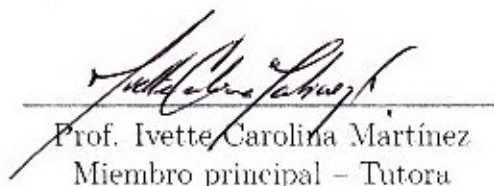
Este Trabajo Especial de Grado ha sido aprobado en nombre de la Universidad Simón Bolívar por el siguiente jurado examinador:



Prof. Masum Nabhan
Presidente



Prof. José Aguilar
Miembro principal (Jurado externo)
Universidad de Los Andes



Prof. Ivette Carolina Martínez
Miembro principal – Tutora

15 de mayo de 2014



UNIVERSIDAD SIMÓN BOLÍVAR
VICERRECTORADO ACADÉMICO
Coordinación de Ciencias de la Computación

ACTA DE VEREDICTO

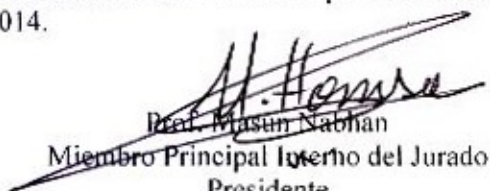
Quienes suscribimos, profesores (as) **Masun Nabhan, Ivette Carolina Martínez y José Aguilar**, miembros del Jurado designado por el Consejo Asesor de la Coordinación Docente de Ciencias de la Computación de la UNIVERSIDAD SIMÓN BOLÍVAR para evaluar el Trabajo de Grado presentado por el estudiante **GÓMEZ CHACÓN, CARLOS ESTEBAN** carné 07-86055 y Cédula de Identidad Nro 17.219.318, bajo el título "MÚSICA INTERACTIVA BASADA EN IMITACIÓN AUTOMÁTICA DE ESTILO". A los fines de cumplir con el requisito legal para optar al Grado Académico de Magister en Ciencias de la Computación, dejan constancia de lo siguiente:

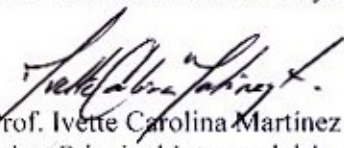
1.- Leído como fue el Trabajo de Grado por cada uno de los miembros del Jurado, éste fijó el día 15 del mes de mayo del año 2014 a las 12:30, para que el autor lo defendiera en forma pública, lo que éste hizo en la Sala 128 Mischa Collar, ubicada en el Edificio MYS, Piso 1, de la Universidad Simón Bolívar, según las siguientes pautas: exposición oral del trabajo por parte del estudiante autor del trabajo, preguntas y comentarios por parte del jurado sobre diversos aspectos conceptuales y metodológicos relacionados con la investigación realizada en el correspondiente trabajo, así como sus resultados, y respuestas del estudiante en cuestión.


2.- Finalizada la defensa pública del Trabajo de Grado, los miembros del Jurado procedimos a deliberar en privado para formular un juicio sobre el Trabajo de Grado y su defensa oral emitiendo el presente veredicto de APROBADO CON MENCION SOBRESALIENTE, apoyándonos en las siguientes razones:

- a.- Tanto en el libro como en la defensa oral, el estudiante mostró dominio en los temas tratados en su investigación, cumpliendo con los requisitos necesarios para optar al grado académico.
- b.- En cuanto al objetivo planteado en su investigación, el mismo fue alcanzado y sobrepasado, ya que además de desarrollar un sistema de música interactiva basado en algoritmos existentes de composición automática, se realizaron mejoras y extensiones. Las mejoras propuestas se evaluaron mediante la realización de experimentos exhaustivos.
- c.- El trabajo tiene todas las potencialidades para ser objeto de publicaciones científicas.

En fe de lo cual se levanta la presente ACTA en Sartenejas, a los 15 días del mes de Mayo del año 2014.


Prof. Masun Nabhan
Miembro Principal Interno del Jurado
Presidente
C.I.: 12130639


Prof. Ivette Carolina Martínez
Miembro Principal Interno del Jurado
Tutora
C.I.: 11071476


Prof. José Aguilar
Miembro Principal Externo del Jurado
C.I.: 8033498

Agradecimientos

Este trabajo contó con la tutoría de la Prof.^a Carolina Martínez. Quiero agradecerla por su ayuda y en especial por su motivación, que fue determinante en la decisión de asumir este proyecto.

Gran parte de las ideas iniciales que desembocaron en este trabajo surgieron de conversaciones que sostenía con el Prof. Eduardo Lecuna acerca de análisis musical automático. Quiero agradecer también a los Profesores Juan Francisco Sans, Adina Izarra y Ernesto Hernández-Novich por sus sugerencias en torno al tema.

Los corales de Bach usados en este trabajo fueron copiados por Margaret Greentree, quien dedicó más de quince años a transcribir los 405 corales de Bach y mantener el sitio jsbchorales.net. Su trabajo ha sido un aporte invaluable para musicólogos, informáticos y melómanos.

Finalmente, quiero darle las gracias a los profesores y estudiantes del Conservatorio Nacional de Música Juan José Landaeta, por muchos años de enseñanza y de amistad.

Caracas, mayo 2014



UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS DE POSTGRADO
COORDINACIÓN DE POSTGRADO EN CIENCIAS DE LA COMPUTACIÓN
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

MÚSICA INTERACTIVA BASADA EN IMITACIÓN AUTOMÁTICA DE ESTILO

Por: Carlos Esteban Gómez Chacón
Carnet No.: 07-86055
Tutora: Ivette Martínez
Mayo 2014

RESUMEN

La composición musical automática es un área activa de investigación hoy en día, en la cual convergen la Inteligencia Artificial y la teoría musical, entre otras disciplinas. Comúnmente, la composición algorítmica es enfocada como un problema de imitación de estilo: generar automáticamente piezas musicales que reproduzcan los rasgos de un estilo dado o de algún cuerpo musical.

Existen una gran variedad de métodos destinados a la imitación automática de estilo. Un método de principal interés son los *sistemas de múltiples puntos de vista*, un formalismo desarrollado por Conklin y Witten basado en modelos de Markov de orden variable. Estos sistemas permiten construir múltiples modelos simultáneos de una superficie musical en base a distintas propiedades abstractas, y combinar las “opiniones” de cada modelo para predecir secuencias musicales, o para generar secuencias musicales nuevas.

En este trabajo, se implementan los sistemas de múltiples puntos de vista para componer secuencias musicales nuevas que imiten una colección musical. En particular, las pruebas se realizan con un cuerpo de melodías tomadas de las 405 armonizaciones corales de Bach. Adicionalmente, el formalismo de los múltiples puntos de vista se extiende para generar secuencias musicales de modo interactivo, en forma de una improvisación guiada por el usuario.

El resultado cuantitativo más importante obtenido es la predicción de las melodías corales con una entropía cruzada de 1,52 *bits/pitch*, lo cual supera el resultado de 1,87 *bits/pitch* obtenido por Conklin y Witten. Finalmente, se implementó un sistema de música interactiva en base a la teoría antes mencionada.

Palabras clave algorithmic composition, multiple viewpoint systems, interactive music, computational creativity

Índice general

1	Introducción	13
2	Marco teórico	21
2.1	Análisis musical	21
2.2	Predicción y generación musical	24
2.3	Sistemas de múltiples puntos de vista	26
2.3.1	Predicción y entropía de música	26
2.3.2	Modelos de contexto	29
2.3.3	Sistemas de múltiples puntos de vista	33
2.4	Tecnología utilizada	44
3	Objetivos del trabajo	47
3.1	Definición del problema	47
3.2	Trabajo relacionado	49
4	Solución propuesta	53
4.1	Arquitectura de la solución	53
4.2	Lectura de la colección musical	54
4.3	Inducción de modelos de contexto	55
4.4	Generación de secuencias	56
4.4.1	Coeficientes de mezcla	57
4.4.2	Funciones de conteo	58
4.5	Composición por lotes	60
4.6	Generación interactiva	61
4.7	Interfaz de usuario	64
4.7.1	Funciones de indización utilizadas	66

5	Experimentos y resultados	69
5.1	Visualización y análisis de puntos de vista	70
5.2	Optimización de los modelos	74
5.2.1	Determinación de los hiperparámetros	75
5.2.2	Predicción de alturas	78
5.2.3	Predicción de alturas y duraciones	82
5.3	Perfil entrópico	84
5.4	Composición de melodías	84
5.5	Generación interactiva	90
6	Conclusiones y recomendaciones	93
6.1	Trabajo futuro	95
	Glosario de términos musicales	107

Índice de cuadros

2.1	Base de datos de contexto	32
2.2	Puntos de vista implementados	42
4.1	Pseudocódigo de la función de generación	56
4.2	Función de indización para el tipo <code>intfref</code>	67
5.1	Rango de los tipos básicos	70
5.2	Escalas mayor y menor	71
5.3	Variables a optimizar	75
5.4	Entropía para distintos valores de r	77
5.5	Predicción del tipo <code>pitch</code> , regla del producto	78
5.6	Predicción del tipo <code>pitch</code> , combinación geométrica	80
5.7	Entropía del sistema $\{\text{pitch} \otimes \text{duration}\}$	80
5.8	Predicción del tipo <code>pitch</code> con modelo de corto plazo	80
5.9	Resultados de Conklin y Witten	81
5.10	Predicción del tipo <code>pitch</code> \otimes <code>duration</code>	83
5.11	Predicción del tipo <code>pitch</code> \otimes <code>duration</code> con modelo de corto plazo	83
5.12	Evaluación humana	87

Índice de figuras

2.1	Generación musical en base a modelos estadísticos	24
2.2	Teoría predictiva de un lenguaje	28
2.3	Arquitectura de múltiples puntos de vista	33
2.4	Tipos producto	35
2.5	Matriz solución de un coral	36
4.1	Arquitectura del sistema	54
4.2	Arquitectura interactiva	61
4.3	Modelo interactivo	63
4.4	Diagrama de flujo de señales	65
4.5	Muestra de la interfaz gráfica	66
5.1	Frecuencia de grados, modo mayor	72
5.2	Frecuencia de grados, modo menor	73
5.3	Frecuencia de grados, orden 1	73
5.4	Frecuencia de intervalos, modo mayor	74
5.5	Determinación de los hiperparámetros	77
5.6	Optimización de la combinación geométrica de puntos de vista	79
5.7	Optimización de la combinación geométrica de modelos	81
5.8	Perfil entrópico	85
5.9	Evaluación humana de las melodías	88
5.10	Melodía 3	89
5.11	Melodía 5	90
6.1	Niveles de significado de la música	97

Capítulo 1

Introducción

Hoy en día, es aceptado que las computadoras igualan o superan a los humanos en tareas como compilar programas, hallar números primos [30], jugar ajedrez [12] e incluso manejar en el tráfico de una ciudad [96]. Existe menos consenso, sin embargo, sobre si las computadoras pueden ser creativas, y superar algún día a los humanos en tareas como componer música, escribir novelas o descubrir teorías matemáticas [20, 98].

Existen dos preguntas relacionadas: *¿Puede una computadora ser creativa?* y *¿Puede una computadora realizar una creación artística bella?*

Para contestar la primera pregunta, es necesario previamente establecer una definición de *creatividad*. Boden [10] plantea una definición operacional de la creatividad: esta se define como la habilidad de generar ideas *nuevas* así como *valiosas* [81]. *Ideas* en este contexto abarca muchos significados: tanto ideas propiamente (conceptos, teorías, narraciones, piezas musicales) como artefactos (imágenes, esculturas, edificios, turbinas de avión). *Valiosas* puede tener también varios sentidos: económico, estético, moral, científico o algún otro [88]. En el contexto de este trabajo se considera suficiente una definición operacional como la anterior; para una definición más completa de creatividad, puede consultarse Newell [65].

La pregunta sobre la creatividad de las computadoras puede ser abordada tanto con argumentos experimentales como filosóficos. Desde el punto de vista experimental, un grupo importante de investigadores en Inteligencia Artificial (AI) se ha dedicado a desarrollar programas que – ellos afirman – exhiben un comportamiento creativo [88]. La metodología que han empleado se basa en utilizar herramientas de la psicología cognitiva para caracterizar de forma empírica los procesos de pensamiento involucrados en la creatividad humana. En base a esta información, los investigadores han desarrollado programas

que simulan o reproducen estos procesos. En su ponencia de aceptación del premio *Outstanding Research* de *IJCAI* en 1999, Hebert Simon resume más de 20 años de literatura perteneciente a esta rama de la AI [88, 33, 77].

Otro enfoque de la AI para lograr la creatividad consiste en elaborar programas cuyos resultados sean creativos, sin necesidad de emular el proceso de pensamiento en sí. Por ejemplo, algunas jugadas de las computadoras que participan en campeonatos mundiales de ajedrez, como *Deep Junior*, son consideradas altamente creativas [12], a pesar de que el mecanismo básico utilizado por estas máquinas es la búsqueda heurística.

Hoy en día, el área de la AI que investiga la creatividad humana y su simulación se denomina *creatividad computacional* [20]. Uno de los principales objetivos de esta área es lograr que las computadoras realicen tareas creativas de forma autónoma como contar cuentos [37], componer música [29], inventar teorías matemáticas [19], dibujar [17] e incluso generar chistes [80].

Además de crear programas que resuelvan tareas autónomas, la creatividad computacional investiga cómo hacer programas que actúen como colaboradores creativos con las personas [20]. Este trabajo está enfocado en esa área, como se verá más adelante.

Existen argumentos encontrados sobre si las computadoras pueden ser “realmente” creativas, en el mismo sentido en que los humanos son creativos. De acuerdo con Boden, responder esta interrogante requiere contestar preguntas aún abiertas sobre “la naturaleza del sentido, o intencionalidad; si una teoría científica de la psicología, o conciencia, es en principio posible; y si una computadora puede alguna vez ser aceptada como parte de la comunidad moral humana” [10]. En todo caso, no se espera una respuesta definitiva a estas preguntas en los próximos cincuenta años [10].

La otra pregunta formulada en este texto es si las computadoras son capaces de generar creaciones artísticas poseedoras de belleza. Douglas Hofstadter, un investigador reconocido en Ciencia Cognitiva, plantea en su libro *Gödel, Escher, Bach* (1979) [46]:

Pregunta: ¿Un programa de computadora compondrá alguna vez música bella?

Especulación: Sí, pero no próximamente. La música es un lenguaje emocional, y antes de que los programas experimenten emociones como las nuestras no habrá manera de conseguir que un programa componga nada dotado de belleza. Puede haber “adulteraciones”, es decir, imitaciones superficiales de la sintaxis de música ya existente, pero, a pesar de lo que se pueda creer a primera vista, en la expresión musical hay mucho más de lo que puede ser capturado por las reglas sintácticas. Falta mucho tiempo para que los programas de computado-

ra compositores de música produzcan nuevos géneros de belleza. Llevaré esta reflexión un poco más allá: pensar –esta sugerencia ha llegado a mis oídos– que pronto podremos ordenar a un modelo de mesa de “caja musical” pre-programada, fabricada en serie y obtenible por veinte dólares mediante envío postal, que haga surgir de sus estériles circuitos composiciones que pudieron haber sido creadas por Chopin o por Bach si hubiesen vivido más tiempo, implica una grotesca y lamentable subestimación de la profundidad del espíritu humano. Para que un programa produzca la música que esos autores produjeron tendría que enfrentar al mundo por sí mismo, afanándose en atravesar el laberinto de la vida y sintiendo cada momento de esa experiencia. Tendría que comprender el gozo y la soledad de una fría noche ventosa, la necesidad de una caricia, la inaccesibilidad de una población distante, el desgarramiento y el consuelo tras la muerte de un ser humano. Tendría que conocer la resignación y el hastío de la vida, el dolor y la desesperación, la determinación y la victoria, la devoción y el temor reverencial. Tendría que haber experimentado la mezcla de elementos opuestos como la esperanza y el miedo, la angustia y el regocijo, la serenidad y la ansiedad. E integrado todo ello como la carne al hueso tendría que tener sentido de la gracia, del humor, del ritmo, y un sentido de lo imprevisto, además, por supuesto, de una exquisita conciencia de la magia de la creación pura. Aquí, y solamente aquí, se encuentran las fuentes de la significación musical. [46]

Sin embargo, el mismo Hofstadter, en textos más recientes [45] afirma que EMI (*Experiments in Musical Intelligence*), un sistema de composición automática desarrollado por David Cope [29], desafía su creencia de que las computadoras actuales no puedan generar música de un alto valor estético.

Por último, en su artículo *Myhill's Thesis: There's More than Computing in Musical Thinking*, Kugel [52] aborda esta pregunta desde el punto de vista de teoría de la computación. Él argumenta que algunos procesos del pensamiento musical humano pertenecen a una clase de procedimientos llamados *procesos de ensayo y error*, los cuales no son computables. Kugel por lo tanto propone que ciertos aspectos del pensamiento musical humano no pueden ser reproducidos por una máquina de Turing.

Composición musical automática

Un problema específico dentro de la creatividad computacional es la *composición musical automática* o *composición algorítmica*. De acuerdo con Papadopoulos, ésta puede definirse como “una secuencia (conjunto) de reglas (instrucciones, operaciones) para resolver un problema (una tarea) [particular] de combinar [en un número finito de pasos] partes musicales (cosas, elementos) en un todo (una composición)” [72].

El presente trabajo está enmarcado dentro del área de la representación simbólica, es decir, las notas y los demás elementos musicales se representan mediante eventos discretos. Un área distinta, mas no totalmente disjunta, es aquella enfocada en el tratamiento de música representada como señales de sonido.

Los primeros trabajos de composición automática a través de una computadora electrónica fueron publicados en 1956 de manera independiente por Hiller e Isaacson (que compusieron la *Suite Illiac*) [44] y Klein y Bolitho (que compusieron la canción *Push Button Bertha*) [1].

En las tres décadas siguientes se realizaron trabajos diversos de composición algorítmica, que en general se caracterizaron por que el autor definía cierto proceso compositivo (por ejemplo, una serie de reglas de construcción melódica), y la computadora producía una salida de acuerdo con este proceso [1].

Las últimas tres décadas (1980–) se han caracterizado por la aplicación de distintos métodos de la AI al problema de composición automática [72, 1]. Hoy en día, existen trabajos recientes de composición algorítmica basados en métodos como gramáticas [90, 89] (incluyendo cadenas de Markov [85, 78, 2]), métodos de satisfacción de restricciones [70], programación lógica [11], algoritmos evolutivos [63] y aprendizaje de máquina [74, 43], así como combinaciones de los métodos anteriores [29, 78, 56]. En [67, 72] los autores presentan una retrospectiva así como el estado del arte de la composición automática.

Existen dos enfoques generales que han sido aplicados para la composición algorítmica. El primero es el enfoque de *modelado del conocimiento*. Bajo este enfoque, el autor codifica explícitamente las reglas de determinado lenguaje musical en un algoritmo, lógica, gramática o algún otro esquema de representación. El segundo es el enfoque de *inducción empírica*, bajo el cual un modelo de cierto estilo musical se construye de forma automática a partir del análisis de composiciones existentes [27].

Existen varios argumentos a favor del segundo enfoque. En primer lugar, un sistema basado en modelado del conocimiento difícilmente puede exhibir un comportamiento creativo, dado que se trata de una entidad programada que sólo puede hacer aquello que sus

programadores le hayan indicado de forma explícita [101]. En segundo lugar, “cualquier sistema lógico de descripción musical poseería demasiadas excepciones, y difícilmente se podría garantizar su completitud; el sistema siempre excluiría algunas piezas válidas” [27, 100].

El análisis musical permite contestar varias preguntas acerca de una pieza: cómo está constituida, cómo se relacionan sus distintos elementos, cómo se relaciona con los rasgos más generales de su estilo, entre otras preguntas. En muchos casos, un análisis de una pieza puede ser visto como una representación simplificada de ella, que omite ciertos detalles para revelar cierto conjunto de propiedades de la pieza. Una pieza, dentro de este esquema, es considerada una superficie musical, en tanto es reflejo de una estructura subyacente.

Conklin y Witten [27] propusieron un formalismo para el análisis musical llamado *sistemas de múltiples puntos de vista*, que permite modelar una superficie musical en base a múltiples representaciones independientes. Dentro de esta teoría, un *tipo* se define como cualquier propiedad abstracta de una o más notas, tales como la altura o duración de una nota, o el intervalo de una nota respecto a la anterior. Un *punto de vista* es un modelo de contexto de una superficie musical en base a un tipo específico. Un *sistema de múltiples puntos de vista* es una teoría que permite predecir secuencias musicales combinando distintos puntos de vista.

Los sistemas de múltiples puntos de vista pueden ser utilizados para la predicción de secuencias musicales, así como para la generación de secuencias nuevas. Conklin [21] discute cómo generar una superficie musical a partir de un modelo estadístico de determinado estilo musical. Esta generación puede realizarse utilizando métodos estadísticos como caminos aleatorios [78], modelos de Markov [79] y muestreo estocástico [101]. Conklin afirma que la distinción tradicional entre modelos analíticos (obtenidos a partir del análisis) y modelos sintéticos (diseñados para la generación musical) es innecesaria, dado que los modelos analíticos pueden ser usados para generar música aplicando cualquiera de estos métodos estadísticos.

Recientemente, varios investigadores han abordado el problema de la generación musical a partir de teorías predictivas basadas en múltiples puntos de vista. Whorley *et al.* [100] desarrollaron un sistema capaz de armonizar melodías de himnos anglicanos, cuyos resultados fueron de calidad variable. Algunas armonizaciones contenían errores de conducción de voces, deficiencias de ritmo armónico y falta de sentido cadencial, mientras que otras resultaban mucho más satisfactorias armónica y estéticamente, no sin contener

algunos errores.

Herremans *et al.* [42] desarrollaron un algoritmo de búsqueda de vecindad variable (*variable neighborhood search* o *VNS*) para componer contrapunto en primera especie a dos voces. Dicho algoritmo utiliza como función objetivo una teoría predictiva basada en puntos de vista verticales. Los autores concluyen que el algoritmo converge a buenas soluciones con muy poco tiempo de cómputo.

Chordia *et al.* [14] utilizaron modelos de múltiples puntos de vista para generar secuencias de tabla (música indostánica). Los autores aseguran que las frases generadas eran novedosas y musicales, aunque dejan para trabajo futuro la evaluación cualitativa detallada a través de una encuesta web. Sostienen además que la generación es un problema significativamente más desafiante que la predicción, dado que la predicción sólo exige determinar un evento dado su contexto, mientras que la generación exige determinar una secuencia completa, lo cual puede divergir rápidamente hacia secuencias inaceptables.

El análisis de múltiples puntos de vista es una instancia específica de una clase más general de estrategias de aprendizaje de máquina, conocidas como *métodos de aprendizaje conjunto* [74]. Ha sido demostrado que este tipo de estrategias puede incrementar la eficacia de los métodos de aprendizaje [32].

Conklin y Witten [27] adicionalmente estructuran la predicción musical como el resultado de la combinación de dos modelos: el *modelo de largo plazo*, que representa el estilo musical inducido de una base de datos amplia de piezas del mismo estilo, y el *modelo de corto plazo*, que representa el estilo de la pieza específica que está siendo predicha.

Música interactiva

En la sección anterior se describió la posibilidad de crear programas de composición automática, que generan composiciones por lotes. Una clase relacionada de programas son los sistemas de *música interactiva*: aquí, la computadora produce una superficie musical que es continuamente influenciada por el usuario en tiempo de ejecución [10].

Ha sido alegado que una desventaja de la composición automática es que los algoritmos pueden ejecutarse infinitamente, y por lo tanto generar una cantidad inmanejable de composiciones [10]. En los sistemas de música interactiva el usuario puede manipular en tiempo real la composición y adaptarla a sus necesidades [82], en otras palabras, el espacio de búsqueda es recorrido de forma guiada por el usuario.

En los sistemas de música interactiva, el usuario puede manipular en tiempo real varios parámetros de alto nivel de la música generada, como la construcción melódica,

la armonía, el tempo, el registro, las dinámicas o el timbre [102]. El valor estético de un sistema de música interactiva no yace solamente en la calidad musical de su salida, sino en la naturaleza de la interacción entre humanos y computador [10]. Parte principal del diseño de un sistema de música interactiva consiste en establecer una correspondencia o *mapping* entre acciones del usuario y los parámetros de la música que se genera [15].

Las interfaces físicas posibles incluyen interfaces tradicionales [73], pantallas táctiles [36], sensores de movimiento y de presión [92], cámaras [103], micrófonos [18] y objetos físicos manipulables [66].

De acuerdo con la clasificación establecida en la sección anterior, la mayoría de los sistemas de música interactiva existentes están basados en modelado del conocimiento [69]. En ellos, el autor diseña un proceso compositivo que condiciona la salida musical del programa. Sin embargo, existen también sistemas de música interactiva que improvisan en base a un modelo estadístico obtenido mediante inducción empírica [69, 4].

Contribución de este trabajo

Este trabajo estudia la generación de secuencias musicales en base al resultado de un proceso de análisis automático de una colección musical. En particular, se realiza una implementación del método de múltiples puntos de vista de Conklin y Witten [27].

Se optimiza un conjunto de modelos para predecir un cuerpo de melodías tomadas de las 405 armonizaciones corales de Bach. La entropía cruzada de la colección respecto al modelo es usada como medida de la capacidad predictiva del modelo. Una hipótesis del trabajo de Conklin y Witten es que los modelos con alta capacidad predictiva tienen también buena capacidad generativa [27].

En la presente investigación se estudia si es factible utilizar el método de múltiples puntos de vista para la composición automática de secuencias musicales. En el trabajo original de Conklin y Witten, los modelos son usados para generar una secuencia musical conociendo su esqueleto rítmico. En otras palabras, el algoritmo compone alturas a partir de un ritmo preestablecido. En el presente trabajo los modelos son usados para componer secuencias de forma integral, generando sus alturas y duraciones.

Adicionalmente, se extiende el formalismo de los múltiples puntos de vista para agregarle la capacidad de generar música de modo interactivo, en forma de una improvisación guiada por el usuario.

Se implementa un sistema de música interactiva basado en esta teoría, con la habilidad de improvisar música al estilo de las melodías de los corales de Bach. Durante la improvi-

sación, distintos parámetros como el registro, la dirección melódica y el ritmo pueden ser manipulados en tiempo real por el usuario.

En los capítulos siguientes se desarrollan los fundamentos teóricos de este trabajo, la definición formal del problema investigado y la arquitectura de la solución. Se presentan los experimentos realizados y finalmente las conclusiones.

Capítulo 2

Marco teórico

El objetivo de este trabajo es la generación de secuencias musicales usando el enfoque de inducción empírica. El primer paso de este proceso es el análisis automático de una colección musical para extraer de ella un modelo estadístico. El segundo paso consiste en generar las secuencias a partir del modelo obtenido.

La primera parte del marco teórico es una revisión de distintos métodos de análisis musical, que constituyen el basamento teórico de este trabajo. La segunda parte de este capítulo explica la relación entre predicción y generación de música, y cómo un modelo predictivo puede ser utilizado para la generación de nueva música. La tercera parte del capítulo está dedicada a exponer el formalismo de los sistemas de múltiples puntos de vista, que son la herramienta principal usada para el análisis en el presente trabajo.

2.1 Análisis musical

De acuerdo con el *New Grove Dictionary of Music and Musicians*, el análisis musical se define como “la resolución de una estructura musical en elementos constituyentes relativamente más simples, y la investigación de las funciones de estos elementos dentro de esa estructura. ... El analista, al igual que el esteta, se ocupa en parte sobre la naturaleza de la obra musical; sobre lo que ella es, abarca o significa; sobre cómo ha llegado a ser; sobre sus efectos o implicaciones; sobre su relevancia o valor para sus receptores”. [84].

Cook agrega que existen dos actos analíticos: el acto de *omisión* y el acto de *relación* ([28] p. 16). Por ejemplo, el análisis armónico mediante números romanos ([28] p. 17) le asigna a cada acorde un número romano que indica cuántos grados por encima de la tónica se encuentra la raíz del acorde. En otras palabras, se indica la *relación* del acorde

respecto a la tónica. Por otra parte, en el análisis de números romanos se indica también la inversión de cada acorde, *omitiendo* especificar cuál es el registro preciso que tiene cada nota del acorde.

Existen distintas corrientes de análisis vigentes, y en esta sección serán explicadas las principales o más relevantes para este trabajo. Puede consultarse [101, 28] para una revisión más completa.

Es importante resaltar dos características del estado actual del análisis como disciplina. En primer lugar, “cualquier análisis (o descripción) de un texto musical asume, a veces de forma implícita o incluso inconsciente, alguna base teórica subyacente” [7]. Distintos métodos de análisis se enfocan en propiedades distintas de un texto musical. En segundo lugar, no existe en la actualidad ningún método completo y universal de análisis que permita explicar satisfactoriamente todas las piezas, ni existen teorías completas de los fenómenos de percepción y cognición musical. Algunos trabajos se han enfocado en probar empíricamente algunas de las teorías existentes o parte de ellas [101], sin embargo, otras no es posible probarlas de forma empírica o implementarlas computacionalmente.

El *análisis Schenkeriano* se comenzó a desarrollar en la década de 1930 y su idea fundamental consiste en reducir de forma recursiva una superficie musical a estructuras armónicas subyacentes más sencillas. El análisis Schenkeriano parte de ciertos axiomas o suposiciones acerca de la naturaleza de la música basados en las prácticas compositivas de los siglos XVIII y XIX, en consecuencia, ciertas piezas son más susceptibles a ser explicadas mediante análisis Schenkeriano que otras. Para una presentación más extensa de esta corriente de análisis, puede consultarse [28]. El análisis Schenkeriano ha sido objeto de varias implementaciones recientes por computador [58, 59].

El libro *Emotion and Meaning in Music* (1956) [61] de Leonard Meyer sentó las bases del *análisis psicológico* musical. Meyer plantea que un estilo musical es un sistema complejo de probabilidades, que rige las expectativas sobre lo que un oyente espera que suceda en un punto cualquiera de una pieza musical. De acuerdo con Meyer, el significado de la música se construye a partir de la satisfacción o negación de estas expectativas [28]. Una teoría reciente basada en estos principios es el modelo de expectativa melódica de *Implicación-Realización* [64] desarrollado por Narmour, que predice en detalle las expectativas o implicaciones producidas por ciertas estructuras melódicas. El modelo de Implicación-Realización ha sido objeto de varios estudios experimentales que han comprobado varios de sus principios de implicación melódica [35, 31], y han sugerido ajustes al modelo en base a la evidencia experimental obtenida.

Otra corriente vigente de análisis es el *análisis comparativo*. De acuerdo con Cook, “si se piensa que el propósito de analizar música es hacer descubrimientos objetivos acerca de la estructura de la música, ... existen dos puntos de partida posibles. La primera posibilidad es divisar una teoría que permita explicar la música en función de alguna clase de principios explícitos de organización. La segunda es adoptar un método comparativo, midiendo distintas piezas una contra la otra; no se necesita una teoría explicativa para este fin, sólo algún tipo de ‘vara de medir’ o criterio para realizar las mediciones” [28]. El análisis comparativo permite por lo tanto responder preguntas sobre qué caracteriza una pieza o conjunto de piezas en comparación con otras, entre otras cosas. Existen varios trabajos computacionales de análisis comparativo [76, 3].

Una teoría reciente de análisis musical basada en aspectos cognitivos es la *Teoría Generativa de la Música Tonal* (en inglés *Generative Theory of Tonal Music* o GTTM) [54]. Esta teoría fue inspirada por la búsqueda de una gramática musical, comparable al modelo de Chomsky de la gramática transformacional del lenguaje natural [54]. El objetivo de la GTTM es “producir una descripción jerárquica y estructural de cualquier pieza de tradición tonal, que corresponda al estado cognitivo final de un oyente experimentado con esa composición” [101].

Temperley desarrolló una teoría computacional cognitiva de percepción musical inspirada parcialmente por la GTTM [94]. En su trabajo, propuso modelos basados en reglas de preferencia que simulan varios procesos cognitivos como la percepción de la estructura métrica, la identificación de la tonalidad y la segmentación de la superficie melódica en frases, entre otros procesos. La mayoría de estos algoritmos de percepción están basados en modelado del conocimiento, y poseen escasa justificación conceptual subyacente [101]. En un trabajo posterior, Temperley reformuló algunos de estos modelos bajo el marco unificado y más general de los métodos Bayesianos [95].

Conklin y Witten [27] propusieron un formalismo para la representación musical denominado *múltiples puntos de vista*, basado en el concepto de tipo algebraico, aplicado específicamente a la música. Un *tipo* es cualquier propiedad de una o más notas, tales como la altura o duración de una nota, el intervalo de una nota respecto al anterior o el contorno melódico de una serie de notas. El análisis de una superficie musical en base a un *tipo* proporciona un *punto de vista* de esta superficie musical. Un *sistema de múltiples puntos de vista* es un análisis de una superficie musical en base a varios tipos y a la interacción entre ellos.

La técnica de múltiples puntos de vista de Conklin y Witten permite crear múltiples

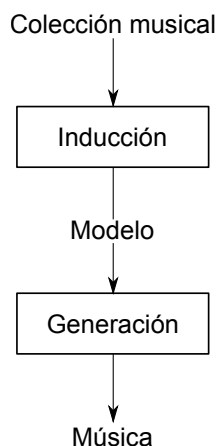


Figura 2.1: Generación musical en base a modelos estadísticos

representaciones simultáneas de una superficie musical, a distintos niveles de abstracción [23]. De acuerdo con la GTTM, la formación de grupos a partir de un flujo musical es parte importante del proceso de percepción musical. Igualmente, varios tipos de análisis presentados en esta sección se basan en extraer estructuras de una superficie musical, y en hallar la relación entre ellas. El formalismo de múltiples puntos de vista es por lo tanto un esquema flexible de representación que sirve como punto de partida para distintas clases de análisis. Ha sido argumentado que la representación basada en múltiples puntos de vista modela de cerca el pensamiento musical humano [72].

Aunque el formalismo inicial desarrollado por Conklin y Witten estaba limitado a música monofónica, posteriormente Conklin introdujo el concepto de puntos de vista verticales para tratar la música polifónica [22] el contrapunto [26]. Varios trabajos recientes han abordado los problemas de análisis y generación musical en base a puntos de vista verticales [100, 42].

2.2 Predicción y generación musical

Como ya ha sido mencionado, uno de los objetivos de este trabajo es la generación musical por el método de inducción empírica, en contraposición al método de modelado del conocimiento. El proceso de generación musical por inducción empírica se esquematiza en la figura 2.1. Un método de inducción es usado para construir un modelo del estilo a partir de un cuerpo de documentos musicales. Posteriormente, un método de generación se utiliza para crear melodías nuevas a partir del modelo.

El proceso de inducción empírica equivale a un proceso de análisis musical, y cualquier

método de análisis musical susceptible a ser implementado computacionalmente puede ser aplicado en este paso. En la sección 2.1 se describieron numerosos métodos de análisis musical, que representan cada uno un posible punto de partida para un sistema que componga música a partir de un proceso de inducción empírica.

El esquema que se plantea conlleva una pregunta fundamental: ¿puede un modelo analítico – cuya función es “decodificar” un texto musical o clasificar sus estructuras – ser usado como modelo sintético – cuya función es “codificar” o generar un texto musical?

Conklin [25] plantea informalmente la pregunta de la siguiente manera:

¿Si tenemos un buen modelo estadístico de un compositor, intérprete, género, etc., cómo podemos usar ese modelo de manera “invertida” para generar música? Este amplio tema trae a colación problemas profundos en modelado estadístico, arquitectura de los modelos, coherencia musical, representación del conocimiento musical y muestreo de modelos estadísticos.

Conklin [21] sostiene que los procesos de análisis musical y generación creativa están altamente interconectados, y que en principio no es necesario hacer la distinción tradicional [83] entre modelos analíticos y modelos sintéticos.

Los modelos analíticos tienen como objeto asignarle alta probabilidad a piezas nuevas dentro de un estilo. Estos modelos pueden ser utilizados, por lo tanto, para guiar un proceso de generación por “ensayo y error” donde las piezas con alta probabilidad se conservan y aquellos con baja probabilidad son descartadas. El proceso de generación ha sido entonces igualado a un problema de búsqueda de un espacio de soluciones utilizando el modelo como función objetivo, o bien de muestreo de un modelo estadístico.

Conklin esboza varios métodos mediante los cuales es posible realizar esta búsqueda, que incluyen caminos aleatorios [78], modelos ocultos de Markov y el algoritmo de Viterbi [79], muestreo estocástico [101] y muestreo basado en patrones [23]. El más sencillo de todos es el camino aleatorio, que consiste en generar un evento aleatoriamente en cada paso de acuerdo con la distribución de probabilidad en ese estado. Dicho método es apropiado para sistemas interactivos que tienen que reaccionar rápidamente; sin embargo, tiene la desventaja de ser una estrategia voraz que tiende a maximizar la probabilidad de los eventos individuales en detrimento de la probabilidad de la secuencia completa.

El camino aleatorio es por lo tanto una aproximación muy simplificada al pensamiento de un compositor, quien puede construir una melodía sin tener que hacerlo “de izquierda a derecha”. La construcción melódica suele estar basada más bien en una estructura jerárquica. La unidad mínima de esta jerarquía es el motivo, varios motivos unidos forman

una semifrase, y así sucesivamente se forman frases, incisos y temas. Se espera que un algoritmo de camino aleatorio sea más exitoso imitando segmentos musicales a pequeña escala, que reproduciendo correctamente estos aspectos de la forma musical.

Varios trabajos recientes aplican la estrategia general de generación musical en base a modelos estadísticos. Herremans *et al.* desarrollaron un algoritmo de búsqueda de vecindad variable (VNS) capaz de generar contrapunto de primera especie, empleando el formalismo de representación de puntos de vista verticales [22, 26]. Herremans llevó a cabo una comparación experimental de la VNS junto con otros dos métodos estadísticos (camino aleatorio y muestreo de Gibbs), y concluyó que la VNS supera a los otros dos métodos. La VNS, que permite modificar sucesivamente de forma aleatoria una melodía en cualquier punto, sin tener que hacerlo de izquierda derecha, debería tener mayor capacidad generativa que el simple camino aleatorio.

Whorley desarrolló un algoritmo de armonización automática a cuatro voces, el cual aplicó a un cuerpo de himnos anglicanos. La generación se realiza a partir de un modelo predictivo basado en múltiples puntos de vista. Como se dijo en el capítulo 1, en algunos casos el algoritmo de Whorley produce resultados aceptables tanto armónica como estéticamente.

2.3 Sistemas de múltiples puntos de vista

En 1950, Shannon propuso un método para estimar la entropía y la redundancia del inglés escrito [86]. Este método es la base de las teorías modernas de compresión de datos, y es el fundamento del formalismo de los *sistemas de múltiples puntos de vista*, un método para la predicción y generación de secuencias musicales desarrollado por Conklin y Witten [27]. En esta sección se presenta la teoría de múltiples puntos de vista, siguiendo de cerca la exposición realizada por los autores. Se describe cómo inducir una teoría predictiva del lenguaje a partir de una colección de documentos musicales, y cómo utilizar esta teoría de forma generativa, para generar secuencias musicales que imiten el estilo de la colección.

2.3.1 Predicción y entropía de música

Un texto de cualquier naturaleza puede contener redundancia. Considérese por ejemplo este fragmento de poesía de Federico García Lorca:

A las cinco de la tarde.

Eran las cinco en punto de la tarde.
Un niño trajo la blanca sábana
a las cinco de la tarde.
Una espuerta de cal ya prevenida
a las cinco de la tarde.
Lo demás era muerte y sólo muerte
a las cinco de la tarde.

Compárese con el siguiente fragmento de poesía de Juan Ramón Jiménez:

La nostalgia, tristísima, arroja
en las almas su amargo silencio,
Y los niños se duermen soñando
con ladrones y lobos hambrientos.
Los jardines se mueren de frío;
en sus largos caminos desiertos
no hay rosales cubiertos de rosas

El primer texto contiene una redundancia evidente, constituida por la aliteración “a las cinco de la tarde”. De manera similar, el lenguaje castellano posee redundancia, y aunque no es siempre tan evidente como en el primer fragmento de poesía, aún así está presente.

Por ejemplo, si usted se encontrara leyendo un poema en una hoja desgastada que dijese: “Quisiera que mi libro fuese / como es el cielo por la no...”, y las últimas tres letras estuviesen borradas, usted podría asumir que la palabra incompleta es *noche*. La palabra también podría ser *norte* o *noble*, pero es razonable decir que esto se cumpliría con menor probabilidad.

En general, un texto contiene redundancia si es posible, utilizando una codificación más eficiente, representar la información en él usando menos caracteres o unidades de información, sin perder la información original.

Volviendo al ejemplo de los poemas, el primer fragmento consta de 241 caracteres, mientras que el segundo, de 234. Si se comprimiera cada fragmento, la representación comprimida del primero seguramente resultaría más corta que aquella del segundo, debido a la mayor redundancia del primer extracto.

En su artículo *Prediction and Entropy of Printed English* (1950) [86], Shannon describe cómo crear una *teoría predictiva* que permita transformar un texto en inglés a una

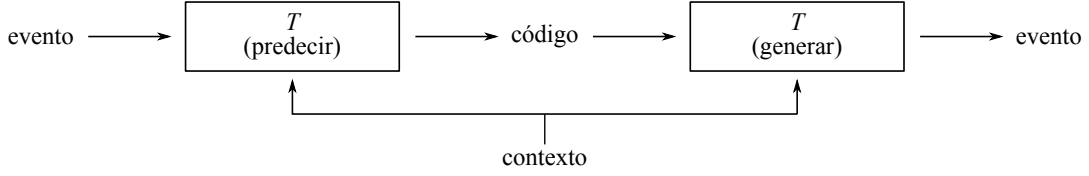


Figura 2.2: Esquema general del principio de compresión en base a una teoría predictiva, según Conklin y Witten [27].

secuencia comprimida de códigos. El proceso es reversible, de manera que empleando la misma teoría predictiva y la secuencia comprimida de códigos, es posible reconstruir el mensaje original.

El proceso descrito se esquematiza en la figura 2.2. Un evento y su contexto (la secuencia de eventos anteriores) le son presentados a una teoría T , que se encuentra en modo “predecir”, para obtener un código comprimido. Al presentarle el mismo contexto y el código comprimido a la teoría T , esta vez en modo “generar”, se obtiene el evento original. Este principio es el fundamento de las teorías modernas de compresión de datos.

Es posible demostrar [51, 60] que si la probabilidad de un evento e , dados un contexto c y una teoría T , es $p_T(e|c)$, el código comprimido no puede tener longitud menor que

$$-\log_2 p_T(e|c) \quad (2.1)$$

Sea $e_1 \cdots e_n$ una subsecuencia de un elemento del lenguaje L . La secuencia $c = e_1 \cdots e_{n-1}$ es el contexto, y $e = e_n$ el evento a ser predicho. La *entropía del lenguaje* L , respecto a una teoría predictiva probabilística T , es la longitud mínima esperada del código de un evento. Puede ser estimada como:

$$\frac{-\sum_{i=1}^n \log_2 p_T(e_i|c_i)}{n} \quad (2.2)$$

donde n es el número de subsecuencias empleadas en la estimación [27].

La cantidad anterior se denomina entropía del lenguaje, entropía cruzada (*cross entropy*) o *log-loss* [8]. La entropía del lenguaje asume que existe un lenguaje el cual se quiere aproximar y un modelo de este lenguaje. La entropía es una medida de incertidumbre, por lo tanto, mientras mejor el modelo, la entropía del lenguaje respecto al modelo será menor.

También es posible pensar en la entropía como una medida de la sorpresa que se obtiene. Si el modelo le asigna probabilidad 1 a una palabra en cierto contexto y acierta,

entonces el nivel de sorpresa es $-\log_2 1 = 0$. Si al contrario el modelo le asignó probabilidad 0 a una palabra en un contexto y la palabra aparece en ese contexto, entonces el nivel de sorpresa es $-\log_2 0 = \infty$, lo cual es generalmente algo muy malo. Generalmente la entropía de una palabra estará entre estos dos extremos.

Como se dijo, la entropía del lenguaje asume que se tiene un lenguaje y un modelo de este lenguaje, es decir $X \sim p(x)$. Desafortunadamente para un proceso empírico no es posible conocer la verdadera distribución de probabilidad $p(x)$, sin embargo, observando instancias, por ejemplo secuencias de un cuerpo musical, es posible producir un modelo de la verdadera distribución. El objetivo al construir este modelo es minimizar la entropía cruzada del lenguaje respecto al modelo que se construye [57].

Leonard Meyer [62] considera que la experiencia musical de un oyente se basa en la generación de expectativas y su posterior satisfacción o inhibición (ver sección 2.1). De acuerdo con Meyer, existe una íntima relación entre este proceso y varios conceptos de teoría de la información. Un estilo musical puede ser visto como un conjunto de probabilidades o una serie de normas que producen expectativas. Un texto musical tendrá baja entropía o cantidad de información si los antecedentes resultan en el consecuente más probable. Por el contrario, cuando un consecuente contradice las expectativas generadas por el antecedente, esta situación tiene una alta entropía o cantidad de información. De acuerdo con Meyer, el significado musical se crea con esta negación de expectativas. Las normas comprendidas en un estilo son para Meyer la condición necesaria para que exista comunicación musical, mientras que el rompimiento de estas normas es la condición suficiente para la comunicación musical.

Aunque Shannon originalmente aplicó este esquema para la predicción de textos, el mismo principio puede ser aplicado para la predicción de música [27, 8, 13]. La representación de los eventos musicales, así como el mecanismo de los múltiples puntos de vista, que amplía notablemente la capacidad de modelar y realizar inferencia sobre un lenguaje musical, serán presentados en esta sección.

Las teorías predictivas que serán desarrolladas en este trabajo serán utilizadas para la generación musical, de acuerdo con la estrategia general descrita en la sección anterior.

2.3.2 Modelos de contexto

Los *modelos de contexto* son una subclase de las gramáticas probabilísticas de estado finito o gramáticas de Markov. También son llamados *modelos de Markov de orden variable*, dado que el número de eventos anteriores usados para predecir un evento depende de la

secuencia de observaciones. En esta sección se describe el algoritmo de inducción de un modelo de contexto, así como el algoritmo de predicción de secuencias a partir de un modelo de contexto. El formalismo de los modelos de contexto servirá como base para presentar en la próxima sección los sistemas de múltiples puntos de vista, que son el tema central de este trabajo.

Definición

Los modelos de contexto son una subclase de las gramáticas probabilísticas de estado finito, también llamadas gramáticas de Markov. La característica particular de los modelos de contexto es que cada variable aleatoria depende de un número de variables que varía según la secuencia de observaciones. Un modelo de contexto tienen tres componentes: 1) una base de datos de secuencias sobre un espacio de eventos, 2) una frecuencia asociada a cada secuencia y 3) un método de inferencia que permite calcular la probabilidad de un evento dado un contexto.

La inferencia a partir de un modelo de contexto se puede realizar de la siguiente manera. La probabilidad condicional $p_T(e|c)$ de un evento e dado un contexto c es el número de ocurrencias de la secuencia $c :: e$ en la base de datos dividida entre el número de ocurrencias del contexto c . Si el contexto c no ha sido nunca antes visto, su frecuencia es 0 y por lo tanto la probabilidad anterior estaría indefinida. Esto se conoce como el problema de la frecuencia cero (*zero-frequency problem*) [104].

El algoritmo de *predicción por reconocimiento parcial* (*Prediction by Partial Match* o PPM) [87, 104] computa la probabilidad de un evento como el resultado de combinar aproximaciones de distinto orden. La probabilidad

$$p_T(e_n|(e_1, \dots, e_{n-1}))$$

se computa realizando una *mezcla*, es decir, una combinación lineal de las cantidades:

$$\begin{aligned} & p_T(e_n|(e_1, \dots, e_{n-1})), \\ & p_T(e_n|(e_2, \dots, e_{n-1})), \\ & \dots, \\ & p_T(e_n|()). \end{aligned} \tag{2.3}$$

En la combinación lineal cada una de las cantidades anteriores se multiplica por un coeficiente c_i directamente proporcional a la longitud del contexto (en otras palabras, los

primeros términos de la enumeración anterior reciben mayor peso). A lo largo de este trabajo dichos coeficientes reciben el nombre de *coeficientes de mezcla*, y en la sección 4.4.1 se presenta el método que fue empleado para calcularlos.

La probabilidad será todavía indefinida cuando el último término de la enumeración valga 0. Esto sucede cuando la secuencia unitaria (e_n) no aparece en la base de datos. Para resolver esto, se agrega un último término $1/|\xi|$ a la combinación lineal. De esta manera la probabilidad de un evento siempre estará definida. El resultado final es una distribución de probabilidad sobre todos los eventos del espacio de eventos ξ , dado un contexto.

Existen dos clases de métodos para combinar las aproximaciones de distinto orden: la mezcla y las *probabilidades de escape*. En 2004 Begleiter *et al.* [8] realizaron una comparación experimental de seis algoritmos de predicción y determinaron que los algoritmos *CTW descompuesto* y PPM consistentemente superaron a los demás algoritmos en diversas tareas de predicción de secuencias. Varios trabajos coinciden en que el mejor método para calcular las probabilidades de escape consiste en el comúnmente llamado “método C”, utilizando una técnica adicional conocida como exclusión [100, 8, 104].

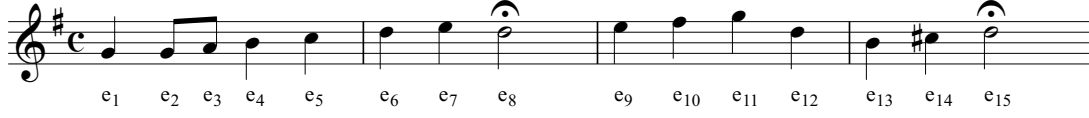
Inducción de modelos de contexto

El algoritmo de inducción de modelos de contexto recibe como entrada una colección de ejemplos y produce como resultado una base de datos de secuencias, en donde a cada una le asocia el número de repeticiones encontradas en la colección.

Para un ejemplo \bar{e}_n de n eventos, el algoritmo de inducción procesa las siguientes n tuplas (contexto, próximo evento):

$$\begin{aligned} &((), e_1), \\ &(e_1, e_2), \\ &((e_1, e_2), e_3), \\ &\dots, \\ &((e_1, \dots, e_{n-1}), e_n). \end{aligned}$$

La inducción procesa una tupla (\bar{e}_{n-1}, e_n) como sigue. Si \bar{e}_n no está en la base de datos, se agrega, se le asigna 1 a su frecuencia, y se procesa recursivamente la secuencia (e_2, \dots, e_n) . Si \bar{e}_n está en la base de datos, se incrementa su frecuencia en 1 y recursivamente se procesa la secuencia (e_2, \dots, e_n) . La operación termina después de procesar la secuencia vacía $()$.



():10	(G):2	(GG):1	(GGA):1
	(A):1	(GA):1	(GAB):1
	(B):1	(AB):1	(ABC):1
	(C):1	(BC):1	(BCD):1
	(D):2	(CD):1	(CDE):1
	(E):2	(DE):1	(DED):1
	(F#):1	(ED):1	(EDE):1
		(EF#):1	(DEF#):1

Cuadro 2.1: Melodía del coral BWV 318 (arriba) y base de datos de contexto después de procesar las diez primeras clases de altura de esta melodía.

Bajo este esquema, el espacio requerido por una tupla $(\overline{e_{n-1}}, e_n)$ es $O(n^2)$, por lo tanto, la base de datos se volvería rápidamente inmanejable. En consecuencia, el tamaño de las secuencias de la base de datos se acota por una constante h , de forma tal que aquellas secuencias de longitud mayor que h no sean agregadas. En otras palabras, para una tupla $(\overline{e_{n-1}}, e_n)$ sólo las secuencias $(e_{n-h+1}, \dots, e_n), (e_{n-h+2}, \dots, e_n), \dots, ()$ son procesadas.

El resultado del algoritmo de inducción se ilustra en la figura 2.1, donde se muestra la melodía del coral BWV 318 así como el estado de la base de datos de secuencias después de haber procesado las 10 primeras clases de altura de esta melodía.

La cantidad $h - 1$ se conoce como el *orden* del modelo. Por ejemplo, en un modelo de Markov de primer orden, $h = 2$.

Modelos de corto y largo plazo

En una pieza musical influyen dos estilos simultáneamente: el estilo general propio del período o compositor y el estilo particular de la pieza. Para la predicción de secuencias musicales, Conklin y Witten representan explícitamente estos dos estilos a través de los modelos de largo y corto plazo.

El modelo de largo plazo representa el estilo general de la pieza y se induce a partir de un cuerpo de documentos musicales pertenecientes al mismo estilo. El modelo de corto plazo es un modelo de la pieza que está siendo predicha, y se construye a partir de la

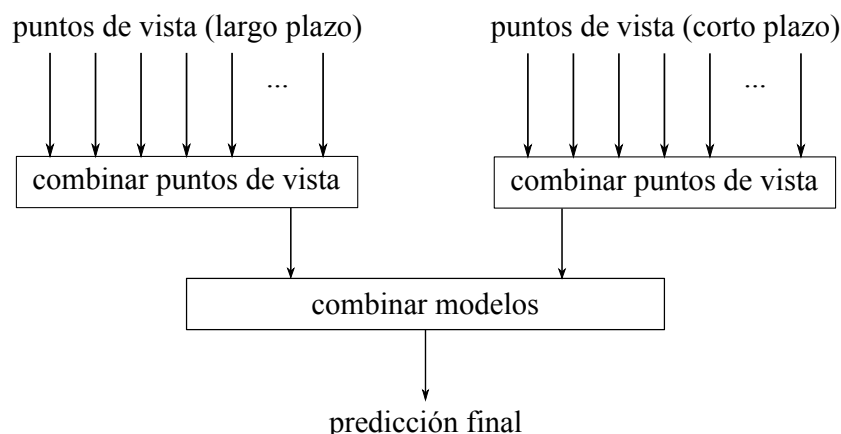


Figura 2.3: Arquitectura de un sistema de múltiples puntos de vista, según Conklin y Witten [27].

pieza que se predice, incorporando en cada paso la secuencia que ha sido predicha.

Más específicamente, sean (c, e) el evento que está siendo predicho y su contexto, el modelo de corto plazo es un modelo del contexto c . El modelo de corto plazo se construye utilizando el mismo proceso de inducción de modelos de contexto descrito, pero a partir de la secuencia que ha sido predicha. El modelo de corto plazo es transitorio, en el sentido de que se descarta una vez terminada la predicción, y dinámico, en el sentido de que cambia a medida que la predicción se realiza. [27]

La probabilidad final de un evento e es una combinación de sus probabilidades de acuerdo con los modelos de corto y largo plazo. La figura 2.3 ilustra el proceso de predicción de un evento en base a modelos de corto y largo plazo.

2.3.3 Sistemas de múltiples puntos de vista

Los modelos de contexto han sido utilizados para problemas de predicción del lenguaje escrito, de clasificación de proteínas y de generación de secuencias musicales, entre otros [8]. Una limitación que poseen es que sólo buscan coincidencias exactas entre secuencias de eventos. La música es un dominio complejo en donde los eventos tienen una estructura interna (altura, duración y tiempo de inicio), y más allá de la superficie musical existen múltiples lenguajes que permiten representar un texto musical.

Ejemplo de esto es que una melodía se puede ver de manera dual como una serie de alturas o una serie de intervalos. Las melodías suelen contener motivos que pueden estar caracterizados por un cierto contorno melódico, más que por una relación exacta de inter-

valos. Las duraciones de las notas pueden obedecer a ciertos criterios puramente rítmicos, pero al mismo tiempo pueden estar relacionadas con la estructura de frase subyacente.

Conklin y Witten introdujeron los sistemas de múltiples puntos de vista como una solución a las limitaciones de los modelos de contexto. Los sistemas de múltiples puntos de vista permiten expresar representaciones simultáneas y distintas de un fenómeno musical, así como interdependencias entre estas representaciones diversas. El formalismo de los múltiples puntos de vista se describe en esta sección.

Tipos derivados

Los puntos de vista permiten utilizar conocimiento previo del dominio de un problema para derivar representaciones variadas de los eventos de una secuencia [27]. Un *tipo* es cualquier propiedad abstracta de un evento, como la clase de altura, el intervalo melódico o la diferencia de tiempo inicial de un evento respecto al anterior. Todo tipo tiene asociada una función Ψ_τ que convierte una secuencia del espacio de eventos original a un elemento de ese tipo. Si τ es un tipo, $[\tau]$ el conjunto de las secuencias sintácticamente válidas formadas por elementos de ese tipo, y $[\tau]^*$ es el conjunto de todas las secuencias posibles de elementos del tipo τ .

Un punto de vista está compuesto por 1) una función parcial $\Psi_\tau : \xi^* \rightarrow [\tau]$ y 2) un modelo de contexto de secuencias en $[\tau]^*$.

Cada punto de vista modela una propiedad particular de los eventos, lo cual origina un nuevo problema. Un punto de vista transforma la secuencia original a una imagen en $[\tau]^*$, en la cual las interacciones entre los tipos básicos pueden haberse perdido. Un sistema tal tendría una escasa posibilidad de representación y de predicción. La solución consiste en modelar las interacciones explícitamente utilizando *puntos de vista enlazados*.

Un *tipo producto* $\tau_1 \otimes \dots \otimes \tau_n$ entre n tipos constituyentes¹ τ_1, \dots, τ_n es a su vez un tipo τ , formado por elementos del producto cartesiano de los tipos constituyentes, es decir, $[\tau] = [\tau_1] \times \dots \times [\tau_n]$. Para un tipo producto $\tau = \tau_1 \otimes \dots \otimes \tau_n$, $\Psi_\tau(\bar{e}_k)$ no está definido si $\Psi_i(\bar{e}_k)$ no está definido para algún $i \in [1 \dots n]$, en caso contrario $\Psi_\tau(\bar{e}_k)$ es una tupla $\langle \Psi_1(\bar{e}_k), \dots, \Psi_n(\bar{e}_k) \rangle$.

El espacio completo de los tipos producto constituye un conjunto parcialmente ordenado bajo la relación de subconjunto entre sus elementos. La figura 2.4 muestra el látice de tal conjunto para los tipos básicos τ_1 , τ_2 y τ_3 . El elemento inferior del látice es el conjunto vacío, y el elemento superior, el producto de los tres tipos. El primer nivel del látice tiene

¹Tipos básicos.

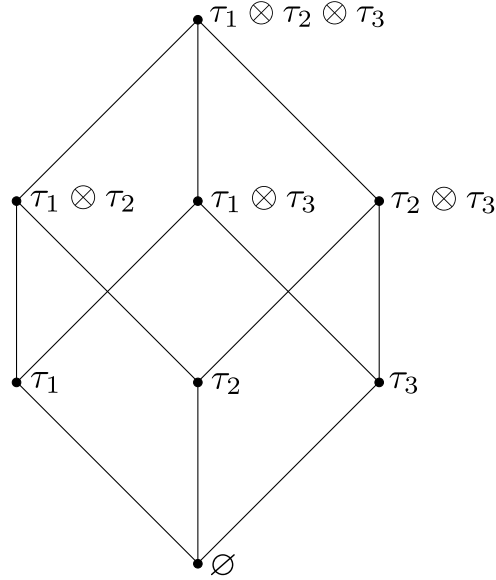
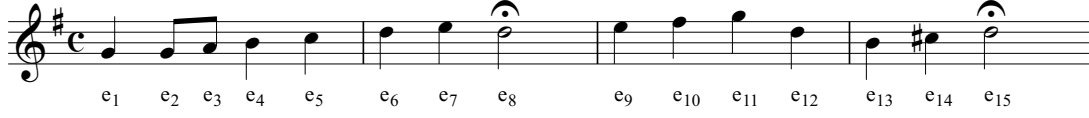


Figura 2.4: Tipos producto formados a partir de tres tipos τ_1 , τ_2 y τ_3 , según Conklin y Witten [27].

todos los tipos primitivos, y el segundo nivel todos los tipos producto de tamaño dos. Un sistema de múltiples puntos de vista puede verse como un conjunto de puntos sobre este látice. Por ejemplo, $\{\tau_1, \tau_2\}$, $\{\tau_1, \tau_1 \otimes \tau_2\}$ o $\{\tau_2, \tau_1 \otimes \tau_2 \otimes \tau_3\}$ son sistemas de múltiples puntos de vista.

En un sistema de n tipos primitivos, el número de sistemas de múltiples puntos de vista primitivos que puede formarse es $O(2^n)$. Si se permiten puntos de vista enlazados de cualquier aridad, el número de sistemas posibles aumenta a $O(n^n)$. Una heurística, tal como el grado de correlación entre los puntos de vista de un sistema, podría ser usada para guiar la búsqueda del mejor conjunto posible de puntos de vista enlazados. La formación constructiva de nuevos modelos durante el proceso de aprendizaje es uno de los problemas más complejos de aprendizaje de secuencias. [27]

Dado un sistema de múltiples puntos de vista, es de interés representar una secuencia junto a las secuencias derivadas, obtenidas cada una mediante un punto de vista. Este conjunto de secuencias derivadas se representa mediante una estructura llamada la *matriz solución*. La matriz solución de n tipos primitivos $\tau_1 \dots \tau_n$ para una secuencia \bar{e}_k es una matriz de $n \times k$, cuya posición (i, j) contiene el valor $\Psi_{\tau_i}(\bar{e}_j)$, o el símbolo \perp si $\Psi_{\tau_i}(\bar{e}_j)$ no está definida. Los tipos producto no necesitan ser representados explícitamente en la matriz, dado que se pueden derivar de las otras filas. La figura 2.5 muestra la matriz



Tipo	Número de evento														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st	0	4	6	8	12	16	20	24	32	36	40	44	48	52	56
pitch	67	67	69	71	72	74	76	74	76	78	79	74	71	73	74
duration	4	2	2	4	4	4	4	8	4	4	4	4	4	4	2
keysig	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
mode	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
timesig	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
fermata	F	F	F	F	F	F	F	V	F	F	F	F	F	F	V
deltast	⊥	0	0	0	0	0	0	0	0	0	0	0	0	0	0
gis221	⊥	4	2	2	4	4	4	4	8	4	4	4	4	4	4
fib	V	F	F	F	F	V	F	F	V	F	F	F	V	F	F
seqint	⊥	2	2	2	1	2	2	-2	2	2	1	-5	-3	2	1
contour	⊥	1	1	1	1	1	1	-1	1	1	1	-1	-1	1	1
referent	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
intfref	0	0	2	4	5	7	9	7	9	11	12	7	4	6	7

Figura 2.5: Las dos primeras frases del coral BWV 318 y su matriz solución para un conjunto de tipos básicos y tipos derivados primitivos.

solución de las primeras dos frases de la melodía del coral BWV 318.

Inferencia usando puntos de vista

Dados un contexto c y un evento e , cada punto de vista perteneciente a un sistema de múltiples puntos de vista debe computar la probabilidad $p_\tau(e|c)$. Esto no se puede hacer de manera directa, porque un punto de vista τ es una distribución de probabilidad sobre $[\tau]^*$ y no sobre ξ^* . Por lo tanto, es necesario primero convertir la cadena $c :: e$ perteneciente a ξ^* a una cadena en $[\tau]^*$. Esta conversión se realiza mediante la función Φ_τ , donde

$\Phi_\tau : \xi^* \rightarrow [\tau]^*$ se define inductivamente como:

$$\begin{aligned} \Phi_\tau(()) &= (), \\ \Phi_\tau(\overline{e_k}) &= \begin{cases} \Phi_\tau(\overline{e_{k-1}}) :: \Psi_\tau(\overline{e_k}) & \Psi_\tau(\overline{e_k}) \text{ definido,} \\ \Phi_\tau(\overline{e_{k-1}}) & \text{en caso contrario} \end{cases} \end{aligned}$$

Para computar la probabilidad de un evento en base a un punto de vista es necesario tomar en cuenta que en general un mismo elemento del punto de vista puede proceder de varios elementos del espacio original. Es decir, la secuencia $\Phi_\tau(c :: e)$ podría representar muchas secuencias además de la secuencia $c :: e$. Por lo tanto, la probabilidad debe ser dividida entre el número de tales secuencias.

El proceso de inducción para los sistemas de múltiples puntos de vista es similar al proceso de inducción de los modelos de contexto tradicionales. Sea (c, e) el ejemplo a procesar. Para todo punto de vista τ , si $\Psi_\tau(c :: e)$ está definida, se añade la secuencia $\Phi_\tau(c :: e)$ a la base de datos del mismo modo que fue descrito en la sección 2.3.2. Es necesario hacer la verificación de que $\Psi_\tau(c :: e)$ esté definida, de lo contrario, la secuencia en $[\tau]^*$ podría ser agregada a la base de datos más de una vez para la misma secuencia de ejemplo. Esto se deduce de la definición de Φ_τ que aparece arriba; si $\Psi_\tau(\overline{e_k})$ no está definida, entonces $\Phi_\tau(\overline{e_k}) = \Phi_\tau(\overline{e_{k-1}})$.

Este proceso de inducción es determinista, es decir, una misma colección musical de entrada siempre producirá el mismo modelo de contexto. El resultado del proceso de inducción, para cada punto de vista, es una base de datos de secuencias.

Desde el punto de vista de la composición musical, el enfoque empleado basado en inducción empírica implica que ninguna regla musical será completamente proporcionada de forma explícita al sistema. Al contrario, el sistema aprende la distribución de los eventos recorriendo la colección musical.

Por ejemplo, una regla bien conocida del contrapunto es que no se permite usar un tritono como intervalo melódico. Esta regla nunca será proporcionada explícitamente al sistema, sino que los tritonos generados serán tan infrecuentes como lo sean en la colección musical.

Inferencia usando múltiples puntos de vista

Un sistema de múltiples puntos de vista puede ilustrarse como una mesa de expertos reunidos para tomar una decisión en torno a cierto problema. Cada experto es especialista

en un área distinta; la decisión final debe combinar la recomendación de cada uno de los expertos.

La figura 2.3 ilustra la arquitectura de un sistema de múltiples puntos de vista constituido por un modelo de largo plazo y uno de corto plazo. La distribución de probabilidad de un modelo se obtiene combinando las distribuciones de probabilidad individuales obtenidas de cada punto de vista que conforma el modelo. La distribución de probabilidad final se obtiene combinando las distribuciones de probabilidad de los modelos de corto y largo plazo.

La combinación de observaciones provenientes de distintos clasificadores ha sido investigada previamente en la literatura. Tax *et al.* [93] compararon las estrategias de promediar y multiplicar la salida los distintos clasificadores para determinar cuál proporciona mejor desempeño. De acuerdo con ellos, la simple operación de promediar las probabilidades a posteriori estimadas por los clasificadores ofrece muy buen desempeño, sin embargo no existe un basamento sólido (Bayesiano) que soporte este método. Por otra parte, multiplicar la salida de los clasificadores individuales es una operación que se deriva de asumir independencia de las probabilidades.

Tax *et al.* concluyen que la regla de combinación del producto es superior a la regla de combinación de la media, pero más sensible al ruido. Por lo tanto recomiendan utilizar el promedio cuando las probabilidades a posteriori no han sido bien estimadas. Cuando los clasificadores han sido entrenados en espacios de características independientes, y arrojan buenos estimados de la probabilidad a posteriori de las clases, se recomienda utilizar la regla de combinación del producto.

Sea $M = m_1, m_2, \dots, m_n$ un conjunto de modelos, la probabilidad combinada de un evento $e \in \xi$ según la *regla de combinación del producto* es:

$$p(e) = \frac{1}{R} \prod_{m \in M} p_m(e)$$

donde R es una constante de normalización.

Pearce *et al.* [74] investigaron reglas para la combinación de múltiples puntos de vista, específicamente dentro del problema de la predicción musical. Ellos propusieron un método multiplicativo basado en la regla de combinación del producto, pero que a diferencia de ésta le asigna un peso a cada distribución de probabilidad. La regla propuesta por ellos

se denomina *regla de combinación geométrica*, y viene dada por:

$$p(e) = \frac{1}{R} \prod_{m \in M} p_m(e)^{\frac{1}{n}}$$

donde R es una constante de normalización. Este es el esquema más básico de la regla de combinación geométrica sin pesos. El esquema con pesos es:

$$p(e) = \frac{1}{R} \prod_{m \in M} p_m(e)^{w_m}$$

donde R es una constante de normalización y los pesos w_m han sido normalizados para que sumen 1.

Un método para calcular los pesos se basa en la entropía de cada uno de los modelos individuales, de forma tal que los modelos con una mayor entropía (es decir una mayor incertidumbre) reciban menor peso. El peso de un modelo m es igual a $w_m = H_{relative}(p_m)^{-b}$. La entropía relativa $H_{relative}(p_m)$ de un modelo viene dada por:

$$H_{relative}(p_m) = \begin{cases} H(p_m)/H_{max}(p_m) & \text{si } H_{max}([m]) > 0 \\ 0 & \text{en caso contrario} \end{cases}$$

donde $[m]$ es el dominio del punto de vista o modelo bajo consideración. El sesgo $b \in \mathcal{Z}$ es un parámetro que produce un sesgo exponencial hacia los modelos con entropía relativa más baja.

Dada una función de masa de probabilidad $p(a \in \mathcal{A}) = P(\mathcal{X} = a)$ de una variable aleatoria \mathcal{X} distribuida sobre un alfabeto discreto \mathcal{A} , la entropía se calcula como:

$$H(p) = H(\mathcal{X}) = - \sum_{a \in \mathcal{A}} p(a) \log_2 p(a)$$

La entropía es máxima cuando todos los símbolos del alfabeto tienen igual probabilidad de ocurrir, es decir $\forall a \in \mathcal{A}, p(a) = \frac{1}{|\mathcal{A}|}$. En este caso

$$H_{max}(p) = H_{max}(\mathcal{A}) = \log_2 |\mathcal{A}|$$

Puntos de vista musicales

En esta sección, se describen los puntos de vista implementados en este trabajo para la predicción de secuencias musicales.

En algunos casos es necesario hacer referencia al valor semántico de un elemento de un punto de vista. Para ello se utiliza el operador $\llbracket \cdot \rrbracket_\tau$, que se lee como el valor semántico asociado a un elemento del tipo τ . En algunos casos el subíndice τ se omite, cuando queda lo suficientemente claro por el contexto.

Tipos básicos En el presente trabajo, las secuencias musicales están limitadas a música monofónica y son representadas como una secuencia de eventos discretos. En otras palabras, las duraciones, tiempos iniciales y todos los demás tipos básicos no admiten valores continuos. El espacio de eventos es:

$$[\text{pitch} \otimes \text{keysig} \otimes \text{mode} \otimes \text{timesig} \otimes \text{fermata} \otimes \text{st} \otimes \text{duration}]$$

donde **pitch** representa la altura del evento, **keysig** es la armadura de clave, **mode** representa el modo (mayor o menor), **timesig** es la indicación de compás, **fermata** es un valor booleano que indica si el evento tiene un calderón, **st** es el tiempo inicial y, **duration** es la duración del evento.

El tipo **mode** no estaba incluido en el trabajo original de Conklin y Witten, por lo cual en ese caso no era posible conocer la tonalidad de una pieza, sino sólo una aproximación proporcionada por la armadura de clave. La colección de entrada usada en el presente trabajo de archivos MusicXML incluye la información del modo de cada pieza, por lo cual se incluye este atributo, que podría provocar una mejora de la capacidad predictiva.

La unidad fundamental de tiempo es la semicorchea; todas las duraciones y tiempos iniciales se expresan como múltiplos de ella. El primer compás, completo o no, de una secuencia siempre comienza en el tiempo cero. Por lo tanto el primer evento de una secuencia puede tener un tiempo inicial mayor que cero, como en el caso de una anacrusa.

Las alturas se representan en semitonos de distancia, donde el do central tiene valor 60. En otras palabras, se sigue la convención MIDI de numeración de las alturas. No se distingue entre sonidos enarmónicos, por ejemplo, tanto el $\text{do}\sharp 4$ como el $\text{re}\flat 4$ tienen valor 61.

Para el tipo **fermata**, $\llbracket V \rrbracket$ significa que el evento tiene un calderón. Incluir los calderones en la representación de los eventos permite expresar fenómenos como la estructura de frase, que en muchos estilos está correlacionada con ciertas duraciones o grados de la escala, por ejemplo, una frase suele comenzar en $\hat{1}$, $\hat{3}$ ó $\hat{5}$, entre otras prácticas.

keysig representa la armadura de clave asociada al evento. La armadura de clave se representa mediante un número entero que puede interpretarse como su posición en el

círculo de quintas. El cero representa que no hay ninguna alteración, el 1, un sostenido, el 2, dos sostenidos, y así sucesivamente. El -1 representa un bemol, el -2 dos bemoles, y así sucesivamente.

El componente **timesig** denota la indicación de compás del evento. Se expresa mediante un número entero que representa la duración del compás, en múltiplos de la unidad básica (semicorchea). Por ejemplo, 12 equivale a una indicación de 3/4, mientras que 16 a una de 4/4.

No existe un tipo explícito para representar los silencios, en cambio, los silencios se representan implícitamente como una diferencia entre el tiempo final de un evento y el tiempo inicial del siguiente, como se detallará en la próxima sección.

El dominio sintáctico de todos los tipos básicos debe incluir todos los valores que se puedan encontrar en la colección de música que se vaya a utilizar. Estos se pueden obtener mediante un simple recorrido de la base de datos, y en el capítulo 5 se detallan los resultados obtenidos para la colección de 350 melodías utilizada. En el cuadro 2.2 se indica el dominio sintáctico de cada tipo.

Tipos derivados Un total de 14 tipos derivados fueron implementados en este trabajo. La figura 2.2 muestra un resumen de los tipos básicos y derivados usados en este trabajo, sin incluir los tipos producto.

La primera columna muestra el nombre simbólico del tipo. La segunda, una descripción informal de su semántica. La tercera columna especifica el conjunto de valores permitidos de elementos del tipo. La cuarta columna indica los tipos básicos de los cuales se deriva el tipo τ , y que por tanto es capaz de predecir.

La parte superior de la tabla muestra los tipos básicos, mientras que la inferior, los tipos derivados.

A continuación se exponen los tipos derivados que fueron implementados, catalogados según el tipo básico del cual se originan.

Tiempo inicial Como se dijo anteriormente, en la representación utilizada todos los eventos tienen altura; no existe una representación explícita para los silencios. En los corales de Bach, los silencios no son frecuentes pero sí ocurren, por lo tanto es necesario que estén contemplados en la representación. Para el tipo **deltast**, $\llbracket v \rrbracket$ significa que la diferencia entre el tiempo inicial del evento y el tiempo final del evento anterior es de v unidades de tiempo.

τ	$\llbracket \cdot \rrbracket$	$[\tau]$	Derivado de
st	tiempo inicial	$\{0, 1, 2, \dots\}$	st
pitch	altura (C4, C \sharp 4, ...)	$\{60, \dots, 79\}$	pitch
duration	duración (semicorchea, corchea, ...)	$\{1, 2, 3, 4, 6, 8, 12, 16\}$	duration
keysig	1 bemol, 1 sostenido, ...	$\{-4, \dots, 4\}$	keysig
mode	mayor o menor	$\{V, F\}$	mode
timesig	3/4 ó 4/4	$\{12, 16\}$	timesig
fermata	con/sin calderón	$\{V, F\}$	fermata
deltast	con/sin silencio	$\{0, 4\}$	st
gis221	diferencia de tiempo inicial	$\{1, \dots, 20\}$	st
fib	comienzo de compás o no	$\{V, F\}$	st
seqint	intervalo en semitonos	$\{-12, \dots, 12\}$	pitch
contour	ascenso / descenso / unísono	$\{-1, 0, 1\}$	pitch
referent	referencia de la pieza	$\{0, \dots, 11\}$	keysig
intfref	intervalo sobre la tónica	$\{0, \dots, 11\}$	pitch, keysig
degree	grado de la escala	$\{0, \dots, 11\}$	pitch, keysig, mode

Cuadro 2.2: Puntos de vista básicos y derivados primitivos implementados, junto con su semántica y su dominio sintáctico.

Para el tipo **gis221**, $\llbracket v \rrbracket$ implica que la diferencia entre el tiempo inicial del evento y el tiempo inicial del evento anterior es de v unidades de tiempo. Nótese que el fenómeno anterior no puede ser capturado por un tipo producto **duration** \otimes **deltast**.

Los eventos también pueden ser caracterizados por su posición en el compás. El tipo **fib** toma un valor booleano, donde $\llbracket V \rrbracket$ si y sólo si el evento se encuentra al comienzo del compás. Resulta útil enlazar este tipo con otras propiedades como la duración o el grado de la escala.

Altura El sistema tonal impone una serie de reglas, explícitas o no, sobre la serie de alturas que pueden conformar una melodía. Las alturas suelen considerarse siempre en relación a cierta escala y no en términos absolutos. Dentro del discurso tonal, una melodía suele encontrarse en una escala determinada, en otras palabras, una melodía exhibe con mayor frecuencia las notas de una escala específica y en menor proporción las notas fuera de esa escala.

Temperley [95] midió la distribución de los grados de la escala en las melodías de la *Essen folksong collection*, una colección de melodías tradicionales europeas, y determinó que los grados que conforman el acorde de tónica, es decir $\hat{1}$, $\hat{3}$ y $\hat{5}$, son los más frecuentes.

Para el tipo **degree**, $\llbracket v \rrbracket$ significa que el intervalo respecto a la tónica de un evento es

v . El tipo **degree** se deriva de **pitch**, **keysig** y **mode**.

En algunas colecciones musicales se incluye información sobre la armadura de clave de cada pieza, pero se desconoce el modo o la escala particular. En esos casos, una aproximación a la tónica es el primer grado de la escala mayor asociada a la armadura de clave. Para el tipo **referent**, $\llbracket v \rrbracket$ significa que el primer grado de la escala mayor asociada a un evento es v . Esta aproximación puede verse como una suerte de “nota pedal” asociada a la melodía, independientemente de que represente o no la tónica. Para el tipo **intfref**, $\llbracket v \rrbracket$ significa que el intervalo del evento respecto a esta “nota pedal” es v .

Una melodía puede caracterizarse alternativamente como una serie de alturas o una serie de intervalos. Narmour [64] enuncia una serie de reglas melódicas inferidas del período de práctica común, y establece, por ejemplo, que intervalos grandes ascendentes suelen estar sucedidos de intervalos pequeños descendentes, y que intervalos pequeños ascendentes suelen estar sucedidos de intervalos pequeños ascendentes. Esto coincide con varias reglas del contrapunto orientadas a mantener el equilibrio melódico [6].

Para el tipo **seqint**, $\llbracket v \rrbracket$ significa que el intervalo de un evento respecto al anterior es de v semitonos. Otra forma muy útil de caracterizar las melodías es a través del contorno melódico. Para el tipo **contour**, $\llbracket \cdot \rrbracket$ toma los valores -1, 0 ó 1, e indica que el intervalo de un evento respecto al anterior es descendente, unísono o ascendente, respectivamente.

Duración En los experimentos de predicción realizados en el trabajo original de Conklin y Witten, la computadora sólo predice la altura de las notas. El esqueleto rítmico del coral que se predice se pasa como entrada al sistema. En cambio, en el presente trabajo se predice tanto la altura como la duración de los eventos. Esto aumenta la complejidad el problema, y hace que la duración sea un tipo de principal importancia para el sistema de múltiples puntos de vista que se construye.

Existen pocos tipos derivados representativos que se pueden obtener a partir de la duración. Una característica general de las melodías de los corales de Bach es la variedad interna de la melodía, en otras palabras, no es correcto dentro del estilo que se repita muchas veces consecutivas una cierta altura o duración. Varios tipos se podrían formular que midiese la variedad de una melodía, donde la variedad se definiría como el número de alturas o duraciones distintas presentes en los últimos k eventos, con k un parámetro. Esto se plantea como trabajo futuro.

En el presente trabajo, sólo se utiliza la duración como un tipo básico, y se construyen varios tipos enlazados con la duración como uno de sus componentes. Por ejemplo, el tipo **duration** \otimes **fermata** puede capturar la tendencia de las notas con un calderón a poseer

un valor más largo.

Calderón La posición en la frase suele estar correlacionada con la altura y duración de las notas. Por ejemplo, la mayoría de las frases empiezan en $\hat{1}$, $\hat{3}$ ó $\hat{5}$, y pocas terminan en $\hat{7}$. En los corales de Bach, la estructura de frase se encuentra anotada implícitamente por los calderones: un calderón es el final de una frase, y se asume que la siguiente frase comienza en el evento siguiente. Además en los corales cada calderón suele marcar una cadencia, que puede ser perfecta, plagal, suspensiva o rota, según su estructura armónica.

Para el tipo **fermata**, $[[V]]$ si y sólo si el evento tiene un calderón. Puede resultar significativo vincular el tipo **fermata** con otros tipos, por ejemplo, el tipo **fermata** \otimes **degree** puede modelar la posición relativa de varios grados de la escala respecto a un calderón.

2.4 Tecnología utilizada

El sistema desarrollado fue implementado en *Haskell* [50], un lenguaje de programación funcional puro. Esto significa que los cálculos en Haskell no tienen efecto de borde, lo cual difiere tangencialmente de la programación imperativa, en la cual el cómputo se hace por la modificación sucesiva del estado de la máquina. El programa se genera con el compilador GHC de Haskell, el cual produce código nativo y optimizado [75].

Para la implementación de la interfaz de usuario y la salida MIDI se utilizó Euterpea [47], un lenguaje de propósito específico basado en Haskell para manipulación musical a alto nivel y síntesis de sonido. Euterpea incluye una biblioteca de programación funcional reactiva [48] para la construcción de interfaces de usuario, que tiene funciones de entrada y salida MIDI en tiempo real.

El módulo de programación funcional reactiva de Euterpea está basado en flechas causales conmutativas (*causal commutative arrows*) [55], un modelo computacional que mejora la eficiencia de las flechas tradicionales, en particular para la programación de sistemas híbridos (es decir, sistemas de componentes tanto continuos como discretos) [48]. Las flechas son una generalización de los *monads*, que tiene una aplicabilidad más amplia que éstos, y en particular resultan adecuadas para expresar cálculos que involucran procesos [49], por ejemplo, en problemas de procesamiento de señales o programación de *dataflow* [55].

La programación funcional reactiva en Euterpea está basada en el concepto de *función*

de señal. Una función de señal es aquella cuyos parámetros son cantidades que varían en el tiempo, por ejemplo, la temperatura proveniente de un sensor o la posición de un robot. Las funciones de señal permiten diseñar procesos interactivos en Euterpea. Tales programas se pueden representar como un diagrama de flujo de señal, que es un gráfico similar a un diagrama de circuito. En la sección 4.7 se muestra un ejemplo de tal diagrama.

Los archivos de entrada fueron representados con el formato *MusicXML* [39], un lenguaje basado en XML para representar la notación musical occidental común. La ventaja de XML sobre el formato MIDI es que permite representar la estructura y la semántica de un documento musical, mientras que MIDI es un formato de bajo nivel diseñado principalmente para la comunicación entre dispositivos de música. Por lo tanto el formato MIDI sólo puede representar información más básica como una secuencia de eventos *note on* y *note off*.

Para la lectura de archivos se utilizó como base el paquete *haxml*, una biblioteca de lectura de archivos XML en Haskell.

Capítulo 3

Objetivos del trabajo

En este capítulo, se define formalmente el problema a investigar en el presente trabajo de grado. Adicionalmente, se ofrece una revisión bibliográfica del trabajo relacionado.

3.1 Definición del problema

El objetivo de este trabajo es desarrollar teorías predictivas que permitan modelar el estilo de un cuerpo de música conocido a priori. Estas teorías pueden ser evaluadas de acuerdo a su capacidad de predecir secuencias en un estilo musical dado. Sin embargo, el verdadero interés consiste en utilizar estas teorías para generar nueva música. En particular, se desea encontrar una teoría con alta capacidad predictiva sobre un lenguaje musical que posea también buena capacidad de generar secuencias musicales nuevas. Se plantea resolver este problema dentro del marco de la teoría de sistemas de múltiples puntos de vista.

El problema planteado puede caracterizarse como un problema de optimización, que consiste en minimizar la entropía del lenguaje investigado de acuerdo con teorías basadas en sistemas de múltiples puntos de vista. Adicionalmente, se desea evaluar de modo subjetivo las teorías construidas según su capacidad de generar ejemplos musicales nuevos, originales, dentro del lenguaje dado.

Por último, este trabajo se propone extender la teoría de múltiples puntos de vista para adecuarla a la generación interactiva de música. En el esquema general a seguir, la melodía generada será producto de un modelo del estilo y de un modelo interactivo, el cual tendrá una serie de parámetros que podrán ser controlados en tiempo real por el usuario.

La culminación de este trabajo implica la realización de los siguientes objetivos espe-

cíficos:

- Para leer la colección de entrada, la cual consiste en las melodías de las 405 armonizaciones corales de Bach [40], se requiere desarrollar una biblioteca de lectura de archivos *MusicXML*.
- Implementar un algoritmo de inducción de un modelo de contexto a partir de una colección de secuencias musicales.
- Implementar un algoritmo de predicción de secuencias musicales en base a una teoría predictiva.
- Definir formalmente varios puntos de vista musicales que potencialmente mejoren la predicción de las secuencias.
- Construir una teoría predictiva, basada en la combinación de múltiples puntos de vista, que minimice la entropía cruzada de una colección de ejemplos musicales. Para esto se debe aplicar un método de optimización. Este método a grandes rasgos deberá generar automáticamente varias teorías predictivas candidatas a partir de un conjunto de entrenamiento, y seleccionar aquellas que minimicen la entropía cruzada del conjunto de prueba.
- Implementar distintos métodos de fusión de puntos de vista. Realizar experimentos para determinar los mejores métodos de fusión, es decir, aquellos que minimicen la entropía cruzada.
- Implementar un algoritmo de generación de nuevas secuencias musicales en base a una teoría predictiva. Evaluar subjetivamente la calidad de las melodías generadas.
- Extender la arquitectura básica de los sistemas de múltiples puntos de vista a través de un modelo interactivo, para hacer posible la generación interactiva de música guiada por el usuario.
- Implementar una interfaz de usuario sencilla que permita probar la generación interactiva de música.

3.2 Trabajo relacionado

Wiggins *et al.* plantean la pregunta de qué constituye una buena melodía, y argumentan que si bien es cierto que las secuencias de más alta probabilidad según un modelo estadístico suelen ser correctas según las reglas musicales, también suelen ser pobres musicalmente [101] (p. 13). Explican que una manera de caracterizar las melodías que son buenas y no meramente musicalmente correctas sería en base a la dinámica que siguen las medidas de teoría de la información, como la *entropía* y el *contenido de información*, dentro de una melodía. Alegan, sin embargo, que este sería un avance posterior, ya que primero es necesario desarrollar un modelo capaz de generar melodías correctas musicalmente. La observación de estos autores es concordante con la tesis de Meyer (ver sección 2.3.1), según la cual diversos procesos fundamentales de la percepción musical como la satisfacción e inhibición de expectativas tienen un paralelo con la entropía.

Cherla *et al.* [13] propusieron un modelo distribuido para la predicción melódica de múltiples puntos de vista, basado en redes neuronales, específicamente en una RBM (*Restricted Boltzmann Machine*). Los autores plantean que las RBM, que han sido previamente aplicadas con éxito al lenguaje natural, tienen varias ventajas sobre los modelos basados en n-gramas, en particular, que escalan linealmente respecto al largo de las secuencias usadas y al número de símbolos de la entrada. Los autores hallan experimentalmente que su técnica supera en capacidad predictiva a los n-gramas, independientemente del tamaño del contexto.

El presente trabajo podría extenderse adoptando una técnica de predicción basada en RBM para compararla directamente con la técnica de n-gramas implementada actualmente. Los autores sólo crean un sistema muy sencillo de dos puntos de vista, así que sería necesario evaluar si el modelo desarrollado por ellos resulta adecuado para un mayor número de puntos de vista. La mejor entropía cruzada obtenida por ellos es de 2,413 *bits/pitch*, la cual es mayor a la de 1,52 *bits/pitch* obtenida en este trabajo.

Whorley *et al.* [100] investigaron el problema de armonizar automáticamente melodías usando múltiples puntos de vista. Los autores señalan que un aspecto fundamental de este problema es determinar el dominio de cada punto de vista, y proponen que este dominio no debe ser fijo, sino que debe determinarse para cada evento que se predice.

Para el caso monofónico, los autores excluyen del dominio de un punto de vista aquellos elementos que resultarían en una nota fuera del espacio de eventos, así como los elementos de los tipos enlazados que resultan contradictorios (por ejemplo, para el tipo `pitch` \otimes `seqint`, cuando existe una contradicción entre la altura y el intervalo). Esta restricción del

dominio se hace con el fin de lograr (en los casos en que es posible) una correspondencia uno-a-uno entre el dominio del punto de vista y el espacio de eventos. En el presente trabajo, este problema se resuelve de otro modo, utilizando las funciones de conteo (ver sección 4.4.2). Las funciones de conteo calculan el número de elementos del espacio de eventos que tienen como imagen un evento del punto de vista. La probabilidad de este evento se divide entre dicho número, de forma que el resultado sea equivalente que si existiera una correspondencia uno-a-uno.

La restricción de los dominios adquiere principal importancia en el caso polifónico: es imprescindible limitar los dominios para que la cantidad de acordes o simultaneidades de notas sea manejable. Whorley *et al.* diseñan tres métodos distintos para modelar la armonía con puntos de vista, utilizando en cada caso reglas distintas para restringir los dominios, y realizan una comparación de desempeño entre los tres. Aunque el énfasis es en la comparación de desempeño, los autores aplicaron el sistema para armonizar melodías de himnos anglicanos, y obtuvieron resultados de calidad variable. Algunas armonizaciones contenían errores de conducción de voces, deficiencias de ritmo armónico y falta de sentido cadencial, mientras que otras resultaban mucho más satisfactorias armónica y estéticamente, no sin contener algunos errores.

Recientemente, Conklin [24] extendió el método de los múltiples puntos de vista con el fin de aplicarlo a la clasificación de melodías. El autor explica que los métodos para la clasificación de música simbólica pueden ser divididos en dos clases: una en donde un vector de características globales se extrae de la pieza y es usado para la clasificación, y otra, como el método de múltiples puntos de vista, donde un modelo del lenguaje se utiliza para generar todos los eventos de una pieza. El autor aplica el método desarrollado a la clasificación de un cuerpo de melodías vascas, y comprueba que el sistema de múltiples puntos de vista supera la eficacia de todos los puntos de vista constituyentes.

Schulze y Van der Merwe [85] desarrollaron un sistema que genera melodías en el mismo estilo de una base de datos de entrada, utilizando modelos de Markov de orden superior y de orden mixto. Los autores alegan que su sistema es capaz de reproducir características musicales de alto nivel del estilo, mientras que otros enfoques sólo son capaces de generar melodías que siguen estrechamente las notas de los ejemplos de entrada.

Pachet [69] desarrolló un sistema de música interactiva llamado *The Continuator*, que improvisa en tiempo real acompañando a un músico, de acuerdo con un estilo inferido de una colección de ejemplos musicales. Pachet ha hecho experimentos con niños de tres a cinco años de edad en los cuales registra y analiza su manera de interactuar con el

sistema, así como varias presentaciones públicas con músicos de jazz que improvisan con el programa [68]. *The Continuator* está basado en técnicas mejoradas de modelos de Markov [5, 53, 97], incluyendo modelos de Markov de longitud variable. Otro sistema de música interactiva basado en inducción del estilo es aquel desarrollado por Assayag y Dubnov [4].

Pachet y Roy estudiaron el problema de *generación dirigible de secuencias de Markov* [71]. Éste problema consiste en generar secuencias de Markov interactivamente de acuerdo a restricciones controladas por el usuario. De acuerdo con Pachet y Roy, los algoritmos tradicionales de generación de secuencias de Markov no se ajustan a este problema porque utilizan una estrategia voraz o *greedy*, que maximiza localmente la probabilidad de cada evento generado. El método de Pachet y Roy permite maximizar la probabilidad de cada trozo generado de la secuencia, y además imponer restricciones adicionales interactivas sobre la generación. La solución de los autores está basada en una reformulación del problema de generación de secuencias de Markov como un problema de satisfacción de restricciones o CSP.

Pazel *et al.* [73] crearon un sistema distribuido de música interactiva en el cual múltiples ejecutantes desarrollan una improvisación colectiva, bajo el control de un director. En una de las modalidades de interacción, el director puede escoger los acordes de la improvisación, mientras que cada participante controla una voz (un instrumento) distinto y las notas de todas las voces se adaptan a los acordes elegidos por el director. Las computadoras pueden estar conectadas remotamente a través de la red.

Otro sistema de música interactiva es *Block Jam* [66]. Éste consiste en un juego de bloques de construcción electrónicos que se pueden conectar entre ellos en forma de estrellas, como piezas de domino. Las secuencias y bifurcaciones que formen los bloques son traducidas a distintas frases y estructuras musicales, que son interpretadas en tiempo real por un sintetizador.

Ciufu [15] ofrece una revisión de sistemas de música interactiva existentes, y discute varias estrategias usadas en la literatura para los problemas de control y generación. En su discusión, distingue entre métodos de generación musical basados en reglas y basados en aprendizaje. Winkler [102] hace una revisión de varias técnicas de mapeo entre movimiento y música usadas en los sistemas de música interactiva. Rowe [82] discute las implicaciones sociales y culturales de los sistemas de música interactiva, y argumenta que una necesidad crítica para la vitalidad y viabilidad de la música en la presente cultura es que un número significativo de personas se envuelva en la creación activa de música, en lugar de nada

más absorber música existente.

Capítulo 4

Solución propuesta

En este capítulo se describen el sistema y los formalismos desarrollados para resolver el problema elegido en este trabajo de grado. El orden del capítulo va de nivel macro a nivel micro, planteando primero la arquitectura general del sistema y luego detallando cada uno de sus componentes.

4.1 Arquitectura de la solución

En esta sección se descompone el problema general planteado en varios subproblemas, a fin de hacer más inteligible la descripción de la solución desarrollada.

La tarea de componer automáticamente en base a modelos de contexto puede dividirse en tres procesos principales: el primer paso consiste en leer una colección musical representada simbólicamente, el segundo en inducir una serie de modelos de contexto a partir de esta colección, y el tercero en generar las secuencias musicales nuevas en base a estos modelos. Este esquema general se ilustra en la figura 4.1.

En este trabajo, existen realmente dos subclases del problema de generar secuencias musicales. La primera es la composición por lotes, que consiste en generar automáticamente secuencias musicales sin intervención del usuario. La segunda es la generación interactiva, que consiste en generar en tiempo real secuencias musicales bajo la influencia de distintos parámetros que pueden variar en función del tiempo.

Por último, encima de este último proceso se ha de desarrollar una interfaz de usuario, que permita controlar la generación interactiva a través de una interfaz gráfica sencilla.

El sistema completo fue desarrollado como un programa en Haskell. Como se verá en la sección 4.7, esto tendrá implicaciones directas sobre el proceso de generación interactiva y

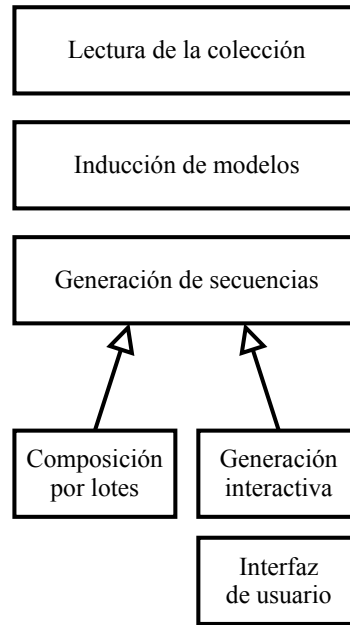


Figura 4.1: Arquitectura del sistema

la interfaz de usuario, ya que el enfoque para construir un programa interactivo en Haskell difiere sustancialmente respecto a los lenguajes imperativos.

En las secciones que siguen se describen por separado cada uno de los subproblemas que han sido identificados. Aunque el énfasis es en la descripción a alto nivel de los procesos desarrollados, cuando se considera pertinente se tratan detalles acerca de la implementación.

4.2 Lectura de la colección musical

Para la representación de los documentos musicales de entrada, se eligió el formato de archivo MusicXML. Esta codificación tiene varias ventajas sobre el formato MIDI para este trabajo: la estructura de compases se representa explícitamente en un archivo MusicXML, lo cual permite conocer el comienzo de cada compás, y esto se usa para leer el punto de vista `fib`. Los calderones también pueden ser representados, y esta información es requerida para el punto de vista `fermata`.

Como colección de entrada se utilizaron 405 melodías luteranas armonizadas por J. S. Bach, transcritas por Margaret Greentree y disponibles en [40]. Dado que este trabajo está limitado a secuencias musicales monódicas, en el proceso de lectura se conservó sólo

la voz de la soprano y se descartaron las otras tres.

En la colección citada todas las piezas se encuentran etiquetadas con su tonalidad (por ejemplo: Do mayor), lo cual representa una información muy valiosa para este trabajo, dado que es posible vincular la tonalidad a distintos tipos básicos para crear puntos de vista que pudieran ser altamente relevantes. Por ejemplo, el tipo `degree`, definido en este trabajo, se deriva de la tonalidad y representa el grado de la escala de una nota. Este tipo sustituye al tipo `intfref` usado en el trabajo de Conklin y Witten, en el cual sólo se disponía de la armadura de clave mas no de la tonalidad (mayor o menor) de las piezas.

Se desarrolló una biblioteca en Haskell para la lectura de archivos *MusicXML*, basada en la biblioteca *haxml* de lectura de archivos XML en Haskell.

4.3 Inducción de modelos de contexto

El proceso de inducción de modelos de contexto fue descrito en la sección 2.3.2. Como se explicó en esa sección, el método de inducción produce una base de datos de secuencias, cada una con una frecuencia asociada. El tipo abstracto de datos apropiado para representar esta estructura es un diccionario (también llamado arreglo asociativo o mapa). Un diccionario es una estructura de datos que almacena un conjunto de pares ($\text{clave}_i, \text{valor}_i$). Conociendo una clave, es posible obtener su valor asociado en el diccionario.

En los experimentos realizados, algunos modelos llegan a comprender más de 40 000 secuencias distintas, por lo tanto, es importante utilizar una implementación de diccionario eficiente.

Existen distintas implementaciones de diccionario en Haskell, basadas en árboles balanceados, tries (árboles de prefijos) y tablas de *hash*. Recientemente, Straka [91] realizó una comparación exhaustiva del desempeño de distintos paquetes de contenedores en Haskell. Al mismo tiempo, desarrolló una nueva implementación de diccionario basada en tablas de *hash*, y determinó que su implementación tiene mejor desempeño general que el diccionario de la biblioteca “estándar” de contenedores de Haskell (`Data.Map`).

En algunas pruebas iniciales realizadas en este trabajo, se determinó que la implementación de Straka resultaba en tiempos de ejecución alrededor de 20 % menores, por lo tanto, se optó por utilizar esta implementación, en lugar de aquella de la biblioteca estándar. La versión actual del programa genera una línea melódica interactivamente utilizando cinco puntos de vista y modelo de orden 5 sin ningún retraso.

```

type Nota = (Int, Float)

-- Genera el siguiente evento de una secuencia.
-- Argumentos:
--   m_lp    Modelo de largo plazo
--   m_cp    Modelo de corto plazo
--   b       True para predecir alturas y duraciones
--           False para predecir alturas
--   k       Indice del evento a ser generado
--   es      Secuencia original (esqueleto de duraciones)
--   xs      Secuencia generada hasta el momento (contexto)
generar :: [Viewpoint] → [Viewpoint] → Bool → Int → [Nota] → [Nota]
        → [Nota]
generar m_lp m_cp b k es xs = xs ++ [x]
  where
    espacio_de_eventos = if b then
      [(x, y) | x ← alturas, y ← duraciones]
    else
      [(x, duracion (es !! k)) | x ← alturas]
    dist_probabilidad = predecir m_lp m_cp espacio_de_eventos
    x = seleccionRuleta dist_probabilidad

```

Cuadro 4.1: Pseudocódigo de la función de generación

4.4 Generación de secuencias

Generar una secuencia consiste en construir de izquierda a derecha la secuencia, seleccionando en cada paso un evento del espacio de eventos. En el trabajo original de Conklin y Witten, el algoritmo de generación recibía como entrada el esqueleto rítmico de la melodía, y el algoritmo sólo debía generar las alturas.

En el actual trabajo, se deben generar tanto las alturas como las duraciones. Para ello, se diseñó una función capaz de generar alturas y duraciones, así como de generar sólo las alturas en base a un esqueleto rítmico. Esto último se utiliza para poder comparar los resultados de este trabajo con aquellos de Conklin y Witten.

La figura 4.1 muestra el pseudocódigo de la función $\text{generar}(m_{lp}, m_{cp}, b, k, es, xs)$, utilizando sintaxis de Haskell. La función que se muestra es una versión simplificada donde algunos detalles de implementación se han omitido. Los parámetros m_{lp} y m_{cp} son los modelos de largo y corto plazo, respectivamente. El parámetro b determina si se deben predecir alturas y duraciones o sólo alturas. El parámetro es es la secuencia original, que se usa para leer la secuencia de duraciones si $b = \text{False}$. El parámetro xs es la secuencia

generada hasta el momento, y la función devuelve esta secuencia con el nuevo evento concatenado al final.

El parámetro k es el índice del evento que está siendo generado. La función *generar* puede ser usada para generar una secuencia a partir de un contexto inicial. Para ello, se le debe pasar como argumento un valor de $k > 0$. Además, el modelo de corto plazo debe haber sido inicializado con el comienzo de la secuencia, lo cual se realiza aplicándole el algoritmo de inducción a la secuencia $s_0 \cdots s_k$.

En el código, *alturas* representa el dominio de las alturas admitidas, y *duraciones* aquel de las duraciones. La función *predecir* recibe una teoría predictiva y una lista de eventos, y devuelve la probabilidad asociada a cada evento según la teoría predictiva. El evento sucesor x se obtiene utilizando el algoritmo de selección de ruleta sobre esta distribución de probabilidad.

Sea $e^1 \cdots e^m$ un conjunto de eventos y $p^1 \cdots p^m$ sus probabilidades asociadas, el algoritmo de selección de ruleta permite obtener un evento e^j al azar de acuerdo con esta distribución de probabilidad. El algoritmo consiste en dividir un segmento de recta unitario en m subsegmentos cada uno del tamaño p^j . Luego, se genera un número aleatorio r con distribución uniforme en el intervalo $[0, 1)$. El valor de retorno del algoritmo es el evento e^j correspondiente al subsegmento de recta en la posición r . Este algoritmo garantiza que cada evento e^j será elegido con probabilidad p^j .

4.4.1 Coeficientes de mezcla

En los modelos de Markov de orden fijo, cada evento de una secuencia depende de un número determinado de eventos anteriores. En cambio, en los modelos de contexto, la probabilidad $p_M(c|e)$ de un evento dado un contexto es el resultado de varias aproximaciones de distinto orden que se combinan para obtener una probabilidad final.

En la sección 2.3.2 se especificó que la probabilidad $p_M(c|e)$ es igual a una suma ponderada de los términos de la ecuación 2.3, en la cual los términos que aparecen más arriba en la ecuación (correspondientes a aproximaciones de mayor longitud) reciben mayor peso. Las aproximaciones de orden 1 (y otras aproximaciones pequeñas) son más fáciles de obtener, por ejemplo, en una situación donde se dispone de pocos datos de entrada, dado que las subsecuencias más pequeñas tendrán mayor tendencia a repetirse. Por otra parte, las aproximaciones de orden superior son más difíciles de obtener y requieren de una mayor cantidad de datos de entrada; sin embargo, aportan un estimado de probabilidad más preciso. Esta es la motivación para el esquema basado en mezcla, en el cual las

aproximaciones de mayor orden reciben un mayor peso en la combinación lineal.

En este trabajo fueron diseñados dos métodos sencillos para el cálculo de los pesos. En el capítulo 5 se realiza un estudio experimental con el fin de determinar el mejor de estos métodos. El primero consiste en una función lineal, donde $\text{lineal}(i)$ representa el peso asociado a los contextos de longitud i . Si n es el orden del modelo:

$$\text{lineal}(i) = \frac{i}{C} \text{ con } 0 \leq i \leq n$$

C es una constante de normalización que se debe elegir de forma que la suma de los pesos sea 1. Debido a la normalización, el valor de los términos no varía con la pendiente de la recta, por lo tanto en este caso se utiliza una pendiente igual a 1.

El segundo método corresponde a una función exponencial o serie geométrica:

$$\text{geométrica}(i) = \frac{r^i}{C} \text{ con } 0 \leq i \leq n$$

r es un parámetro del sistema, que será determinado en la fase de experimentos. De nuevo C es una constante de normalización.

4.4.2 Funciones de conteo

Un punto de vista τ comprende una función Φ_τ que transforma una secuencia del espacio de eventos original, es decir ξ^* , a $[\tau]^*$. Esta función, en general, es de muchos a uno, en otras palabras, múltiples secuencias del espacio de eventos original pueden tener la misma imagen.

Cuando se utiliza un punto de vista para calcular la probabilidad de una secuencia $c' :: e'$ en $[\tau]^*$, esta probabilidad está definida sobre τ y no sobre el espacio de eventos original.

Para transformar de vuelta la probabilidad al espacio de eventos ξ , es necesario dividirla entre el número de posibles secuencias en ξ^* que tienen como imagen la secuencia $c' :: e'$. Dada una secuencia c en ξ^* , la función de conteo $\text{conteo}_\tau(c' :: e')$ se define como el número de eventos e en ξ tales que $\Phi_\tau(c :: e) = c' :: e'$.

Para la exposición que sigue, es necesario recordar la definición de un evento:

`(pitch, keysig, mode, timesig, fermata, st, duration)`

Para el tipo `pitch`, un elemento $\llbracket v \rrbracket$ puede provenir de cualquier evento que tenga al-

tura v y una duración cualquiera dentro del dominio de las duraciones. El campo **fermata** también puede tener cualquier valor admitido (V o F). Los tipos **keysig**, **mode** y **timesig** se fijan como parte del dominio de los eventos de una pieza, por lo tanto no deben ser tomados en cuenta aquí. En la implementación actual no se generan silencios, por lo tanto el campo **st** también es fijo al momento de predecir un evento particular.

En base a lo anterior, el número de posibles eventos asociados a un mismo valor del tipo **pitch** es $|\text{duration}| \otimes |\text{fermata}|$. Por lo tanto

$$\text{conteo}_{\text{pitch}}(\bar{s}_n) = |\text{duration}| \cdot |\text{fermata}|$$

Las funciones de conteo de los tipos **duration** y **fermata** se determinan de manera análoga:

$$\text{conteo}_{\text{duration}}(\bar{s}_n) = |\text{pitch}| \cdot |\text{fermata}|$$

$$\text{conteo}_{\text{fermata}}(\bar{s}_n) = |\text{pitch}| \cdot |\text{duration}|$$

Para el tipo **gis221**, $\llbracket v \rrbracket$ implica que la diferencia entre el tiempo inicial del evento y el tiempo inicial del evento anterior es de v unidades de tiempo. En ausencia de silencios, el tipo **gis221** determina por lo tanto la duración del evento anterior. La función de conteo para el tipo **gis221** es por lo tanto

$$\text{conteo}_{\text{gis221}}(\bar{s}_n) = |\text{pitch}| \cdot |\text{duration}| \cdot |\text{fermata}|$$

Para el tipo **seqint**, $\llbracket v \rrbracket$ significa que el intervalo melódico respecto a la nota anterior es de v semitonos. El tipo **seqint** determina por lo tanto unequivocamente la altura de una nota. La función de conteo del tipo **seqint** es por lo tanto

$$\text{conteo}_{\text{seqint}}(\bar{s}_n) = |\text{duration}| \cdot |\text{fermata}|$$

Para el tipo **contour**, $\llbracket v \rrbracket$ significa que el intervalo respecto al evento anterior es ascendente, descendente o unísono. Si $\llbracket v \rrbracket$ es un unísono, la altura queda determinada porque tiene que ser igual a la nota anterior. Si $\llbracket v \rrbracket$ es ascendente, la altura puede ser cualquiera mayor a la nota anterior, y si es descendente, cualquier altura menor. No es posible conocer la altura anterior, dado que tal como están diseñadas actualmente las funciones de conteo reciben la secuencia de elementos de punto de vista y no la secuencia de eventos original. Sin embargo es posible realizar una aproximación. Si las alturas están

uniformemente distribuidas en el dominio (aquí $[60, 79]$), en el caso promedio el valor de la altura estará en la mitad de este rango. Por lo tanto la función de conteo puede definirse como:

$$\text{conteo}_{\text{contour}}(c' :: e') = \begin{cases} 1 & \text{si } \llbracket e' \rrbracket \text{ es un unísono} \\ |altura| \div 2 & \text{si } \llbracket e' \rrbracket \text{ es ascendente} \\ |altura| - |altura| \div 2 & \text{si } \llbracket e' \rrbracket \text{ es descendente} \end{cases}$$

Para el tipo **intfref**, $\llbracket v \rrbracket$ implica que el intervalo respecto a la “tónica” es de v semitonos (como se explicó en la sección 2.3.3, esta “tónica” es como una nota pedal derivada de la armadura de clave y no necesariamente representa la tonalidad, dado que no se conoce si el modo es mayor o menor). El dominio de las alturas es $[60, 79]$, lo que equivale a las notas C4 a G5. Por lo tanto si $\llbracket v \rrbracket$ está entre 0 y 7 ambos inclusive, puede provenir de dos octavas distintas, mientras que si está entre 8 y 11 ambos inclusive, sólo puede provenir de la octava inferior. La función de conteo se define por lo tanto como:

$$\text{conteo}_{\text{intfref}}(c' :: e') = \begin{cases} 2 \cdot |duration| \cdot |fermata| & \text{si } 0 \leq \llbracket e' \rrbracket \leq 7 \\ |duration| \cdot |fermata| & \text{en caso contrario} \end{cases}$$

La función de conteo para el tipo **degree** se define de manera análoga:

$$\text{conteo}_{\text{degree}}(c' :: e') = \begin{cases} 2 \cdot |duration| \cdot |fermata| & \text{si } 0 \leq \llbracket e' \rrbracket \leq 7 \\ |duration| \cdot |fermata| & \text{en caso contrario} \end{cases}$$

4.5 Composición por lotes

El sistema desarrollado tiene la capacidad de generar secuencias musicales nuevas por lotes, es decir, sin intervención del usuario. El modelo de corto plazo recibe como contexto inicial la primera frase de una melodía de la colección (la primera frase está comprendida por las notas hasta el primer calderón inclusive), y el resto de las notas son generadas por el sistema.

La composición se realiza mediante la función de generación descrita en la sección 4.4. Actualmente, el algoritmo genera melodías de longitud fija N (en los experimentos realizados $N = 50$). El problema de determinar automáticamente el final de una melodía no es tratado en este trabajo.

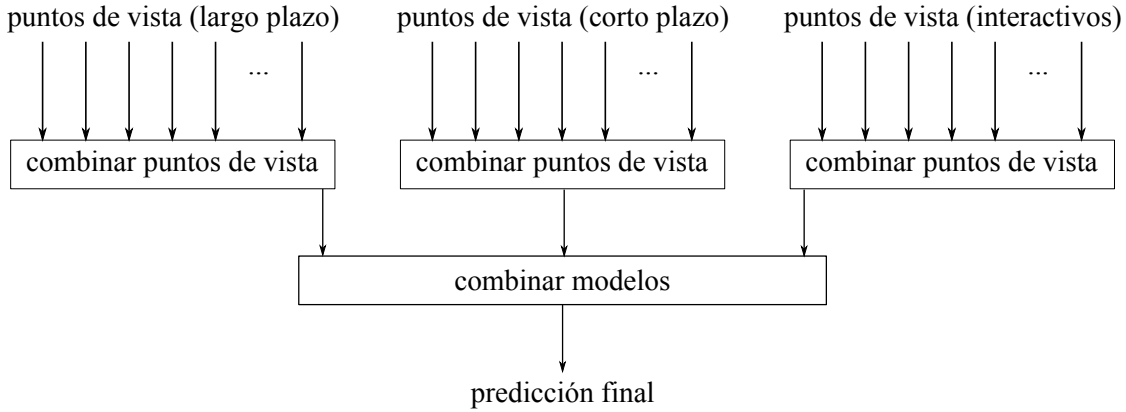


Figura 4.2: Arquitectura de un sistema de múltiples puntos de vista con modelo interactivo

4.6 Generación interactiva

De acuerdo con el método de predicción de Conklin y Witten, la distribución de probabilidad sobre ξ dado un contexto se calcula combinando las distribuciones de probabilidad obtenidas por los modelos de largo y corto plazo. Este esquema fue descrito en el capítulo 2 (ver figura 2.3).

Para la generación interactiva, es necesario combinar los modelos de largo y corto plazo con un tercer modelo que arroje una distribución de probabilidad que sea producto de la interacción con el usuario. En el marco de este trabajo, este modelo recibe el nombre de *modelo interactivo*. Este nuevo esquema de predicción se ilustra en la figura 4.2.

Para obtener la probabilidad de un evento dado un contexto, el modelo interactivo combina las predicciones de varios puntos de vista. Sin embargo, aquí un punto de vista τ se define como 1) una función parcial $\Psi_\tau : \xi^* \rightarrow [\tau]$, 2) una función de indización índice $\tau : [\tau]^* \rightarrow [l_\tau, u_\tau] \subseteq \mathcal{Z}$ y 3) una señal de entrada del usuario $x_\tau \in [0, 1] \subseteq \mathcal{R}$.

Típicamente, en un sistema interactivo la interfaz estará constituida por un conjunto de *sliders* o cualesquiera controles de interfaz cuyo estado se pueda expresar mediante un valor escalar. Por lo tanto, para poder establecer alguna ley o correspondencia entre la posición de un control y los elementos de un punto de vista, es necesario mapear cada elemento del punto de vista a una coordenada en la recta real. Éste es el objetivo de la función de indización.

Por ejemplo, para un punto de vista hipotético *acorde*, la función de indización podría mapear los acordes más disonantes a números negativos cada vez menores, y los acordes más consonantes a números positivos cada vez mayores, de esta manera, a través de un *slider* sería posible controlar la tensión de los acordes generados.

Existen dos enfoques generales para el diseño de funciones de indización. El primero consiste en diseñar la función manualmente, mediante algún tipo de fórmula o reglas de decisión que le asignen una posición en la recta real a los elementos del dominio. Por ejemplo, para el caso de los acordes, la disonancia se puede medir en función de los intervalos verticales formados entre cada par de notas del acorde. Los intervalos verticales, de acuerdo con la teoría musical, se pueden clasificar fácilmente entre disonantes (por ejemplo, las segundas, el tritono y las séptimas) y consonantes (las terceras, quintas, sextas y octavas). De esta manera es posible estimar el grado de disonancia de un acorde en función del número de intervalos disonantes que contenga.

El otro enfoque consiste en utilizar algoritmos de agrupamiento automático para descubrir los segmentos más representativos estadísticamente de puntos de vista y agruparlos automáticamente en un conjunto de clases [23, 22]. Para cada punto de vista, estas clases pueden ser ordenadas manualmente según el criterio del diseñador del sistema interactivo, para así obtener una función que le asigna un orden numérico a segmentos de punto de vista. Un problema de este enfoque es que durante la generación pueden surgir segmentos que nunca han sido vistos antes, por lo tanto la técnica de agrupamiento debe ser capaz de clasificar correctamente elementos nunca vistos anteriormente.

Para cada punto de vista τ , dado un valor real $x_\tau \in [0, 1]$ que representa el valor actual de la señal proveniente de la interfaz de usuario, se debe calcular la probabilidad de $(c :: e) \in [\tau]^*$.

El valor *delta* representa la diferencia entre el valor de la señal y el valor de la función de indización para la secuencia $c :: e$, este último normalizado al intervalo $[0, 1]$.

$$delta = \frac{\text{índice}_\tau(c :: e) - l_\tau}{u_\tau - l_\tau} - x_\tau$$

En la fórmula anterior, u_τ y l_τ son los límites superior e inferior, respectivamente, de la función de indización.

Para calcular la probabilidad $c :: e$, se utiliza una distribución normal con media 0 y varianza σ . El área bajo la curva se delimita por el intervalo $[-z^*, z^*]$ correspondiente al valor crítico p . Este intervalo se divide en $u_\tau - l_\tau$ intervalos iguales:

$$i = \frac{2z^*}{u_\tau - l_\tau}$$

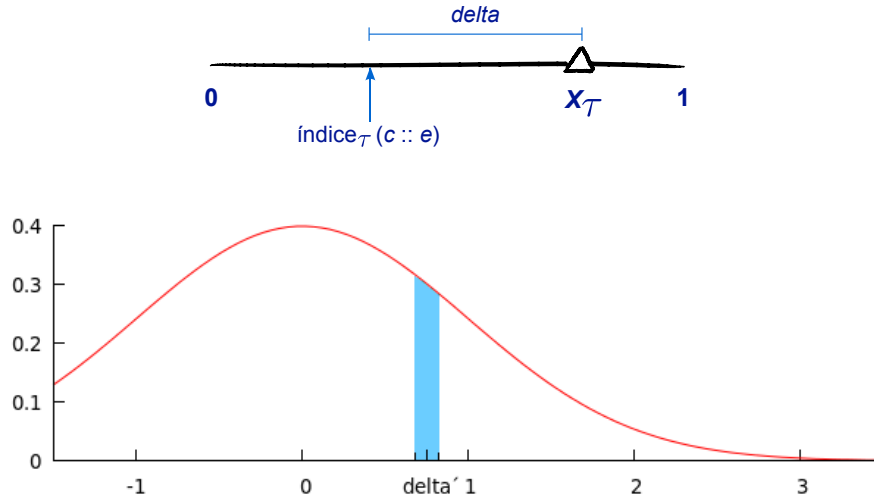


Figura 4.3: Cálculo de la probabilidad de una secuencia $c :: e$ de tipo τ según el modelo interactivo.

El valor $delta$ se transforma del rango $[-1, 1]$ al rango $[-z^*, z^*]$

$$delta' = delta \cdot z^*$$

La probabilidad $c :: e$ es igual al área de la región bajo la curva delimitada por el intervalo $[delta' - \frac{i}{2}, delta' + \frac{i}{2}]$:

$$p_{\tau}(c :: e) = \text{cdf}(delta' + \frac{i}{2}) - \text{cdf}(delta' - \frac{i}{2})$$

En la expresión anterior, la función cdf es la función de distribución acumulada de una distribución normal con media 0 y varianza σ :

$$\text{cdf}(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2\sigma^2}} dt$$

La varianza σ es un parámetro del sistema.

La figura 4.3 ilustra el cálculo de la probabilidad de una secuencia $c :: e$ de tipo τ según el modelo interactivo. La probabilidad de un evento según este modelo sigue una distribución normal alrededor de la posición del slider indicada por el usuario. Esto es conveniente porque de esta manera el modelo interactivo no arroja un solo evento, sino que favorece a una región de eventos alrededor de la posición correspondiente del slider. Esto debería producir mayor variedad en la música generada.

4.7 Interfaz de usuario

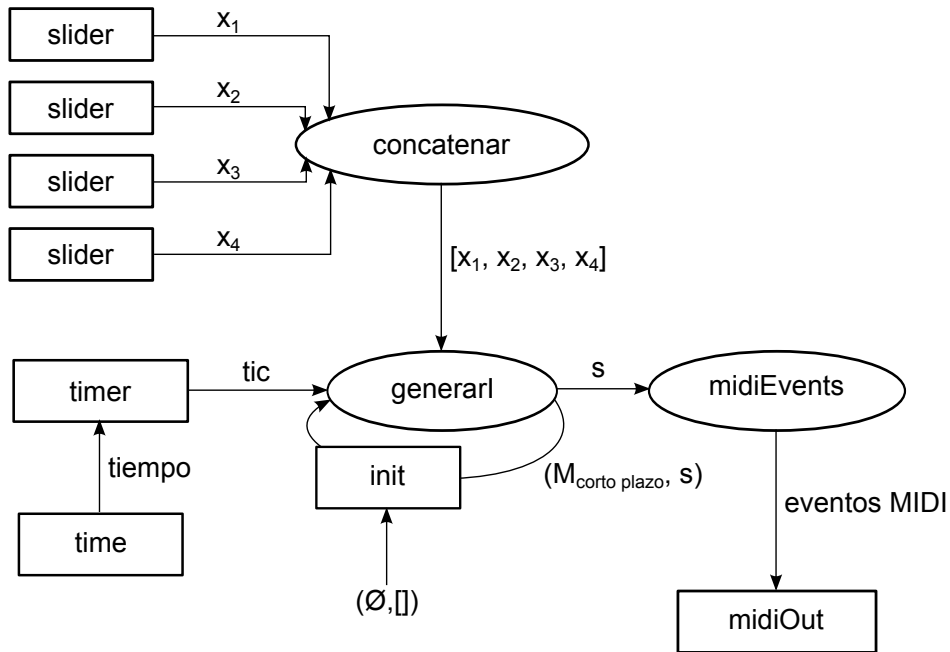
Se desarrolló una interfaz de usuario sencilla para controlar la generación interactiva basada en el lenguaje Euterpea. Como fue explicado en la sección 2.4, este lenguaje permite construir interfaces de usuario en base a funciones de señal. El diagrama de flujo de señales de la interfaz de usuario se muestra en la figura 4.4, junto con la implementación en Haskell usando sintaxis de flechas (el código mostrado es una versión simplificada donde algunos detalles de implementación se han omitido).

La función **time** genera una señal que representa el tiempo actual, y se utiliza como entrada a la función **timer**, la cual genera una señal positiva cada cierto intervalo de tiempo. En la implementación actual la frecuencia de este temporizador es de 1/8 segundos, lo cual corresponde a la duración de una semicorchea si se asume un tempo de $\text{♩} = 120$. En otras palabras, el mínimo intervalo de tiempo entre una nota y la siguiente a efectos de la generación interactiva es de una semicorchea.

El sistema actual utiliza cuatro puntos de vista interactivos: **pitch**, **seqint**, **intfref** y **duration**. La interfaz de usuario, que consta de cuatro sliders para controlar estos puntos de vista, puede verse en la figura 4.5. Los sliders sólo admiten posiciones enteras en el rango $[-4, 4]$. El 0 es la posición neutra, es decir, cuando el slider tiene ese valor no influye en la generación.

Cada slider tiene una señal de salida x_τ que corresponde a su posición actual. Estas señales son concatenadas y pasadas a la función **generarI**, que se encarga de generar la siguiente nota de la improvisación. La función *generarI* está implementada en base a la función *generar* presentada anteriormente (sección 4.4), pero tiene un mecanismo adicional que le permite “dormir” tantos ciclos como dure la última nota generada, para despertar justo un ciclo antes de que tenga que sonar la próxima nota.

La función *generarI* produce las notas una por una, y actualiza el modelo de corto plazo en cada paso. Esta función forma parte de un ciclo recursivo (especificado mediante el combinador *rec* en el código), ya que la secuencia generada y el modelo de corto plazo actualizado son a su vez entrada para la misma función en el siguiente instante de tiempo. La función **init** provee la entrada inicial (la secuencia vacía y el modelo de corto plazo vacío) para la primera llamada a la función *generarI*. La secuencia producida por la función *generarI* también es enviada a **midiEvents**, función que traduce de una secuencia de eventos a una secuencia MIDI. Por último, los eventos MIDI son enviados a la función **midiOut** de Euterpea, que se encarga de realizar la salida MIDI.



```

ui :: [Viewpoint] → [Viewpoint] → [Viewpoint] → UISF () ()
ui m_lp m_cp m_in = proc _ → do
  x1 ← title "pitch"    $ hiSlider 1 (-4, 4) 0 < ()
  x2 ← title "seqint"   $ hiSlider 1 (-4, 4) 0 < ()
  x3 ← title "intfref"  $ hiSlider 1 (-4, 4) 0 < ()
  x4 ← title "duration" $ hiSlider 1 (-4, 4) 0 < ()
  t ← time < ()
  tic ← timer < (t, 0.125)
  rec s ← init (m_cp, []) <
    if (isJust tic) then
      (generarInteractivo m_lp m_in [x1,x2,x3,x4]) s
    else s
  let eventos_midi = if (isJust tic) then Just (midiEvents s)
    else Nothing
  midiOut < eventos_midi

```

Figura 4.4: Diagrama de flujo de señales de la interfaz de usuario, y código Haskell correspondiente utilizando la sintaxis *do* de flechas.

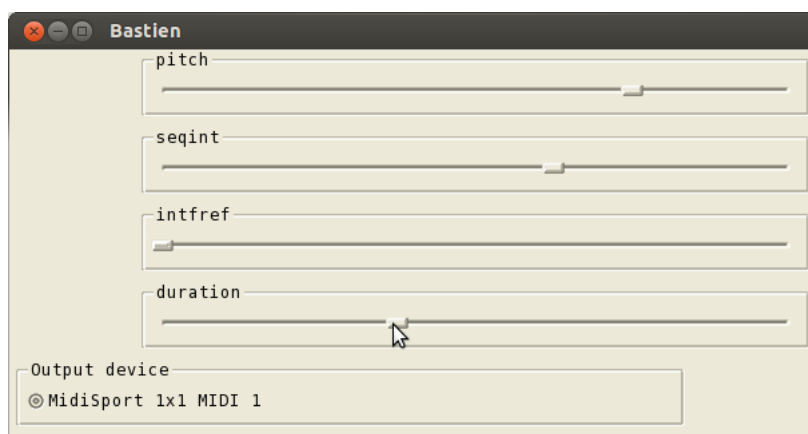


Figura 4.5: Muestra de la interfaz gráfica del sistema

4.7.1 Funciones de indización utilizadas

En el caso de los puntos de vista interactivos **pitch**, **seqint** y **duration**, la función de indización es la función identidad: la altura, el intervalo y la duración simplemente estarán en correspondencia directa con la posición del slider correspondiente.

En el caso del punto de vista **intfref**, en lugar de ordenar los grados de la escala de menor a mayor, tiene más sentido ordenarlos según su función dentro de la escala. En música existe una organización jerárquica entre los grados de la escala, donde la tónica o primer grado es el centro tonal, y los demás grados tienen una función alrededor de ésta. Los grados $\hat{5}$ y $\hat{7}$ contribuyen altamente a reforzar la tonalidad, ya que tienen una función de dominante, es decir, producen expectativa de resolver a la tónica. Otros grados importantes son el $\hat{2}$, que también produce tensión hacia la tónica al ser adyacente, y el $\hat{4}$ que es la subdominante. Por otra parte, los grados alterados no pertenecen a la estructura básica de la escala y tienden a debilitarla. La función de indización se diseñó según este criterio, asignándole un valor de 11 al grado principal (la tónica) y un valor de 0 al grado más extraño de la escala. La tabla completa de valores se muestra en la figura 4.2.

v	grado	$\text{índice}_{\text{intfref}}(v)$
0	1	11
1	#1 / b2	3
2	2	8
3	#2 / b3	0
4	3	6
5	4	7
6	#4 / b5	4
7	5	9
8	#5 / b6	1
9	6	5
10	#6 / b7	2
11	7	10

Cuadro 4.2: Función de indización para el tipo `intfref`.

Capítulo 5

Experimentos y resultados

El objetivo de los experimentos es, en primer lugar, realizar una búsqueda de la mejor teoría predictiva, recorriendo el espacio de sistemas de múltiples puntos de vista y seleccionando aquel que arroje la mejor capacidad predictiva sobre la colección musical.

Una vez encontrado, este modelo puede ser utilizado para generar melodías nuevas, y también para generar las notas durante la improvisación interactiva. Como se verá en este capítulo, no siempre sucede que la mejor teoría predictiva sea la mejor teoría generativa. En los experimentos realizados, la mejor teoría predictiva tuvo que ser adaptada para que generara melodías satisfactorias.

El primer experimento reportado es un experimento preliminar, que consistió en visualizar las distribuciones de probabilidad de varios puntos de vista. Esto sirve principalmente para validar que el sistema esté construyendo las teorías predictivas correctamente, pero además es de interés como herramienta para el análisis musical, por ejemplo dentro de la musicología.

El segundo experimento tuvo como propósito optimizar la capacidad predictiva de los modelos, buscando exhaustivamente el mejor conjunto de puntos de vista.

El tercer experimento consistió en visualizar el perfil entrópico del proceso de predicción de una melodía coral, lo cual permite explicar y apoyar el método de predicción. Para la predicción de las melodías, en esta prueba se utilizó el mejor modelo obtenido en el experimento anterior.

El cuarto experimento radicó en componer automáticamente un lote de melodías corales, utilizando el mejor modelo obtenido anteriormente. Estas melodías fueron sometidas a una evaluación humana.

El último experimento consistió en probar la generación interactiva de música.

Tipo	Rango
pitch	$\{60, \dots, 81\}$
duration	$\{1, 2, 3, 4, 6, 8, 12, 16\}$
keysig	$\{-4, \dots, 4\}$
mode	$\{F, V\}$
timesig	$\{12, 16\}$
fermata	$\{F, V\}$
deltast	$\{0, 4, 8, 12\}$

Cuadro 5.1: Rangos de entrada de los tipos básicos, obtenidos mediante un recorrido de la colección.

Colección utilizada

Para los experimentos se empleó la colección de 405 armonizaciones corales de J. S. Bach, publicada por Margaret Greentree y disponible en [40] (ver sección 4.2). Sólo se utilizó la voz de la soprano y las demás fueron descartadas, en otras palabras sólo se utilizó la melodía original. Estas melodías no fueron compuestas por Bach, sino que se remontan a fuentes anteriores.

Por razones de eficiencia, los corales fueron representados utilizando la semicorchea como unidad mínima de tiempo. Se descartaron algunos corales que contenían notas con duraciones extremas, cambios de métrica o acompañamiento instrumental. Por lo tanto, el número definitivo de corales usados fue 350.

El cuadro 5.1 muestra los rangos de entrada de los tipos básicos, extraídos mediante un recorrido de la colección.

5.1 Visualización y análisis de puntos de vista

Un modelo de contexto expresa la probabilidad de que una nota cualquiera aparezca en una pieza musical, dado un contexto. Un punto de vista es potencialmente una herramienta muy útil para el análisis musical, dado que permite definir formalmente una propiedad cualquiera de una superficie musical, y observar su distribución de probabilidad dentro de una pieza, o dentro de un cuerpo de obras musicales.

El primer experimento de este trabajo estuvo enfocado en definir varios puntos de vista y visualizar su distribución de probabilidad en la colección de melodías corales. La principal utilidad de este experimento dentro de este trabajo es validar que el sistema esté construyendo correctamente las teorías predictivas, observando las distribuciones de

degree	escala mayor	escala menor
0	1	1
1	#1 / b2	#1 / b2
2	2	2
3	#2 / b3	3
4	3	b3/b4
5	4	4
6	#4 / b5	#4 / b5
7	5	5
8	#5 / b6	6
9	6	b6/b7
10	#6 / b7	7
11	7	b7

Cuadro 5.2: Equivalencia entre el tipo **degree** y los grados de la escala mayor y menor.

probabilidad calculadas por el sistema a partir de la colección musical.

Se analizaron por separado las melodías mayores y menores de la colección, dado que las escalas mayor y menor tienen distintas configuraciones interválicas. La escala menor es igual a una escala mayor donde el tercer grado, sexto y séptimo han sido rebajados. La tabla 5.2 muestra la configuración de las escalas mayor y menor según la teoría musical. La primera columna representa los grados de la escala en número de semitonos, que es la representación usada por el tipo **degree**. La segunda columna muestra el grado correspondiente de la escala mayor, y la tercera columna aquel de la escala menor.

Para este experimento se eligió analizar los tipos **degree** y **seqint**. El primero representa el grado de la escala asociado a una nota, es decir no representa la altura absoluta de la nota sino su altura relativa a la tonalidad de la pieza. El tipo **seqint** representa el intervalo respecto a la nota anterior, y también es fundamental dentro del discurso musical. De hecho, una melodía puede ser vista alternativamente como una secuencia de alturas o una secuencia de intervalos.

La figura 5.1 muestra la distribución de probabilidad de orden 0 del tipo **degree** en las melodías mayores. Se observa que las notas de la escala tienen una probabilidad mucho más alta que aquellas fuera de la escala. La nota más frecuente es la tónica, seguida de los grados $\hat{3}$ y $\hat{2}$. Los siguientes grados con más frecuencia son el $\hat{5}$ (la dominante) y el $\hat{4}$ (la subdominante). Esta distribución difiere levemente de aquella de la *Essen Folksong Collection* graficada por Temperley [95], donde los grados del acorde de tónica ($\hat{1}$, $\hat{3}$ y $\hat{5}$) eran los más frecuentes, y el segundo grado tenía una probabilidad menor que éstos.

La figura 5.2 presenta la distribución de probabilidad de orden 0 del tipo **degree** en

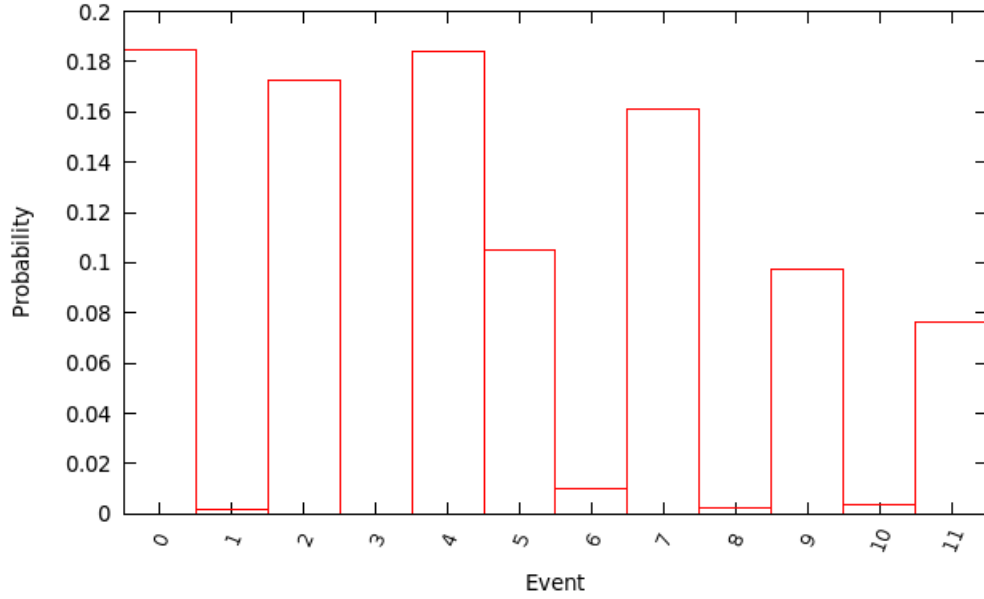


Figura 5.1: Distribución de probabilidad de orden 0 para el tipo **degree** en las melodías mayores.

las melodías menores. La distribución refleja la estructura de la escala menor, donde los grados $\hat{3}$, $\hat{6}$ y $\hat{7}$ de la escala menor predominan ahora en lugar de los correspondientes de la escala mayor.

La figura 5.3 muestra la distribución de probabilidad de orden 1 del tipo **degree** en las melodías mayores. La gráfica tiene la forma de una matriz, donde las filas corresponden al contexto y las columnas al siguiente evento. La casilla (c, e) de la matriz indica la probabilidad de que el grado c vaya seguido del grado e .

Esta gráfica es útil porque, a diferencia de la primera que mostraba la probabilidad de eventos individuales, muestra la probabilidad de la transición de un evento a otro. En otras palabras, la gráfica de orden 0 sólo proporciona información sobre la frecuencia de los eventos mas no sobre su posición relativa en la secuencia, información que sí se puede inferir de la gráfica de orden 1.

La fila 0 corresponde al grado $\hat{1}$, y se observa en esta fila que los sucesores más frecuentes de este grado son el $\hat{2}$, el $\hat{1}$ y el $\hat{7}$. Esto indica que en la mayoría de los casos la tónica está sucedida de un unísono o de un movimiento por grado conjunto. En la fila 2 se pueden ver los sucesores del grado $\hat{2}$, que en su mayoría es seguido por los grados $\hat{3}$, $\hat{2}$ y $\hat{1}$ (de nuevo se observa una tendencia hacia los grados adyacentes). El sucesor más frecuente en la fila 5 (correspondiente al grado $\hat{4}$) es el grado $\hat{3}$, lo cual corresponde a

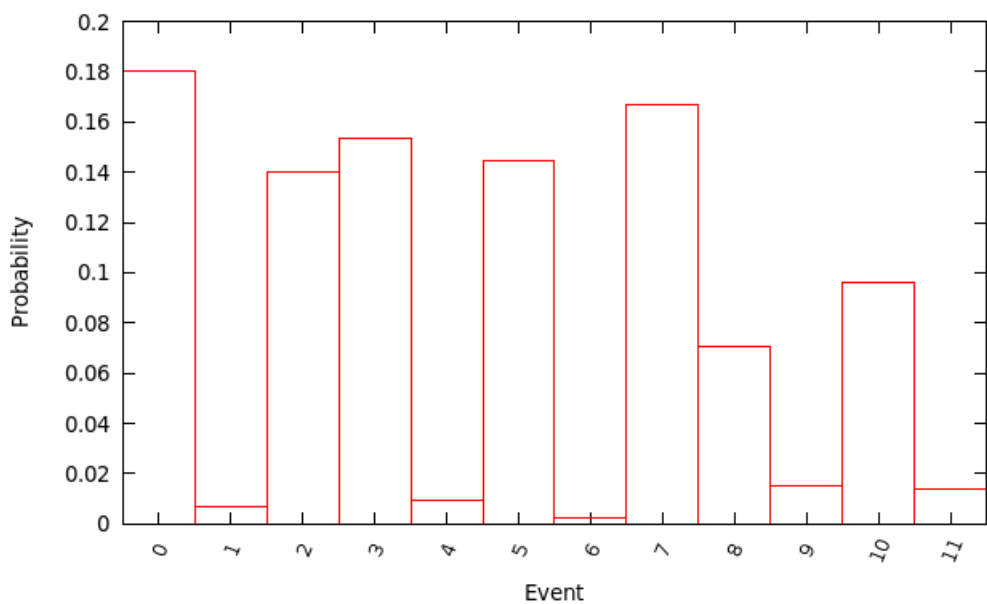


Figura 5.2: Distribución de probabilidad de orden 0 para el tipo **degree** en las melodías menores.

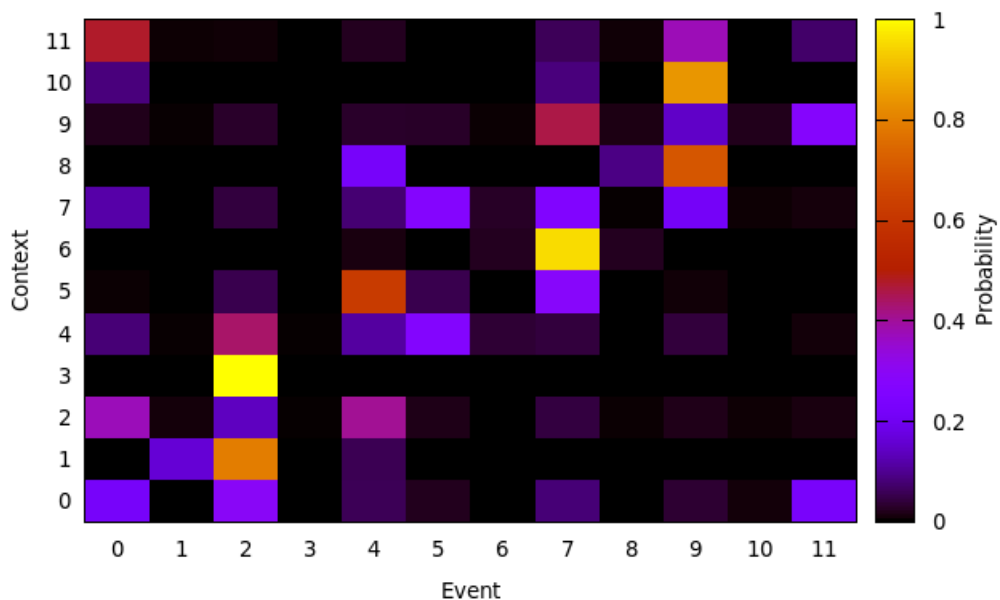


Figura 5.3: Distribución de probabilidad de orden 1 para el tipo **degree** en las melodías mayores.

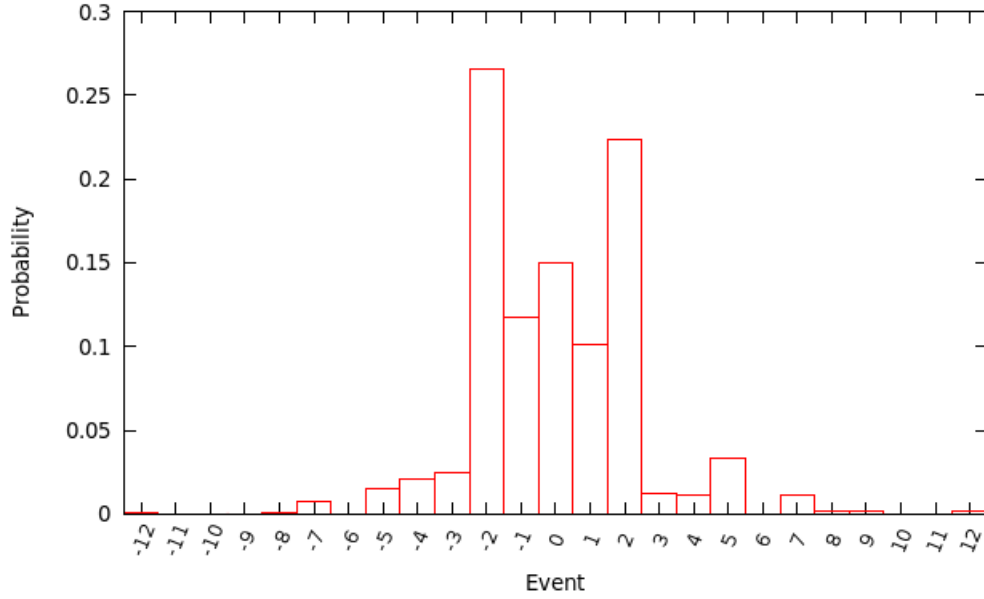


Figura 5.4: Distribución de probabilidad de orden 0 para el tipo **seqint** en las melodías mayores.

un movimiento de segunda menor descendente (por ejemplo la resolución de la séptima en la cadencia V-I). Este tipo de movimiento es característico en las reglas armónicas de conducción de voces. En la fila 11 se observa que la mayoría de las veces el grado $\hat{7}$ es seguido de la tónica, lo cual también es consistente con la armonía convencional.

Por último, en la gráfica 5.4 se presenta la distribución de orden 0 del punto de vista **seqint**, en la colección de melodías mayores. Las segundas mayores ascendentes y descendentes son los intervalos más comunes, seguidas de las segundas menores y el unísono. Los demás intervalos tienen una frecuencia considerablemente menor, lo cual indica que en las melodías corales predomina el movimiento por grado conjunto.

5.2 Optimización de los modelos

La colección fue particionada aleatoriamente en un conjunto de entrenamiento de 340 corales y un conjunto de prueba de 10 corales. La búsqueda del mejor modelo fue planteada como un problema de optimización, que consistió en usar el conjunto de entrenamiento para inducir un conjunto de modelos, y seleccionar aquel que minimizara la entropía cruzada del conjunto de prueba. Los valores de la entropía cruzada que se reportan en los experimentos que siguen son el promedio sobre los diez corales del conjunto de prueba.

Variable	Rango
Orden de los modelos	$l \in \mathcal{N}^+$
Método de mezcla	lineal exponencial $r \in [1 \dots] \subseteq \mathcal{R}$
Método de combinación	Regla del producto Regla geométrica $b \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 16, 32\}$
Conjunto de largo plazo	$X_{lp} \subseteq C, 1 \leq X_{lp} \leq 5$
Conjunto de corto plazo	$X_{cp} \subseteq C, 1 \leq X_{cp} \leq 5$

$$C = \{\text{pitch}, \text{duration}, \text{seqint}, \text{contour}, \text{degree}, \text{seqint} \otimes \text{gis221}, \\ \text{degree} \otimes \text{seqint}, \text{degree} \otimes \text{fib}, \text{keysig} \otimes \text{pitch}, \text{mode} \otimes \text{seqint}, \\ \text{mode} \otimes \text{degree}, \text{pitch} \otimes \text{duration}\}$$

Cuadro 5.3: Variables a optimizar y sus rangos

Existían múltiples variables a optimizar: el orden de los modelos, el método de mezcla, el método de combinación de puntos de vista, el método de combinación de modelos y el conjunto de puntos de vista de largo plazo y de corto plazo. La tabla 5.3 resume las variables a optimizar.

El método de mezcla lineal no tiene ningún parámetro, mientras que la mezcla exponencial tiene un parámetro r que se debe determinar. En cuanto a los métodos de combinación, la regla del producto no tiene ningún parámetro, pero la regla geométrica tiene un parámetro b que ha de explorarse.

El conjunto C es el conjunto de todos los puntos de vista que fueron implementados (excluyendo aquellos que no predicen los tipos `pitch` ni `duration` y por lo tanto no aportan a la predicción). Tanto para el modelo de largo plazo como el de corto plazo, se consideraron todos los subconjuntos posibles de C de tamaño 1 a 5. El número de tales conjuntos es $12 \cdot 11 \cdot 10 \cdot 9 \cdot 8 = 12^5 = 95040$.

Con cada sistema candidato, se creó un modelo a partir del conjunto de entrenamiento utilizando el algoritmo de inducción. La función objetivo fue la entropía cruzada de la colección de prueba respecto al modelo, cuya fórmula puede verse en la ecuación 2.2.

5.2.1 Determinación de los hiperparámetros

El primer paso de la optimización consistió en determinar el valor de los hiperparámetros, es decir, el orden de los modelos y el método de mezcla. Para ello se diseñó un modelo de un único punto de vista `pitch` \otimes `duration`, que se utilizó para predecir el tipo `pitch` \otimes

duration. Se midió la entropía cruzada utilizando modelos de orden $l = 0 \dots 30$ y el método de mezcla lineal así como exponencial con valores de $r = 1, 1.1, 1.2, \dots, 2$.

La estrategia consistió, por lo tanto, en determinar primero los hiperparámetros con un sistema de puntos de vista trivial, midiendo la capacidad de predecir el tipo **pitch** \otimes **duration**, que es el tipo más complejo que se va a predecir en este trabajo. Sin duda esta no es la única estrategia posible, y dado que existe interacción entre el conjunto de puntos de vista y los hiperparámetros, sería deseable explorar los dos simultáneamente.

En el futuro un método de búsqueda heurística podría ser implementado, por ejemplo, un algoritmo genético, que recorra el espacio de parámetros más eficientemente¹. Sin embargo, actualmente los resultados obtenidos con el procedimiento de optimización realizado son satisfactorios, en el sentido que la entropía cruzada obtenida para la predicción de alturas mejora significativamente los trabajos previos conocidos, como se verá en la siguiente subsección.

La figura 5.5 muestra la entropía cruzada obtenida en función del orden del modelo² y el método de combinación, incluyendo distintos valores del parámetro r . Cada curva corresponde a un valor particular de r . En la gráfica los valores menores en el eje Y son mejores.

En general, hasta $l = 7$ los modelos de orden mayor producen una mejora, pero a partir de este punto seguir aumentando el orden no produce suficientes ventajas. Esta observación coincide con varios trabajos anteriores [8, 16]. Los valores de r alrededor de 1.6 producen una menor entropía cruzada, y ésta empeora al alejarse hacia arriba o hacia abajo de este valor. La mezcla lineal supera al método exponencial para modelos de orden 6 y superior, sin embargo, antes de este punto el método exponencial resulta mejor.

Se decidió usar un modelo de orden 5. Aunque los modelos de orden 6 y 7 tenían una menor entropía cruzada, cada unidad en que se incrementa el orden de los modelos resulta en un aumento considerable del tamaño de la base de datos de secuencias. Se consideró por lo tanto que no se justificaba incrementar el orden más allá de 5.

El cuadro 5.4 muestra el resultado asociado a cada valor de r para $l = 5$. Se eligió la constante $r = 1.6$, que fue el valor óptimo obtenido con el modelo de orden 5.

¹Como se vio en la sección 2.3.3, la complejidad de encontrar el mejor sistema de múltiples puntos de vista primitivos y enlazados a partir de n tipos primitivos es $O(n^n)$. Whorley *et al.* [100] propusieron una heurística basada en *stepwise selection* para recorrer eficientemente el espacio de posibles sistemas de múltiples puntos de vista, sin embargo su método no garantiza el óptimo global.

²Dado que el comportamiento no mejoraba después de $l = 16$, la gráfica sólo muestra hasta esa abscisa.

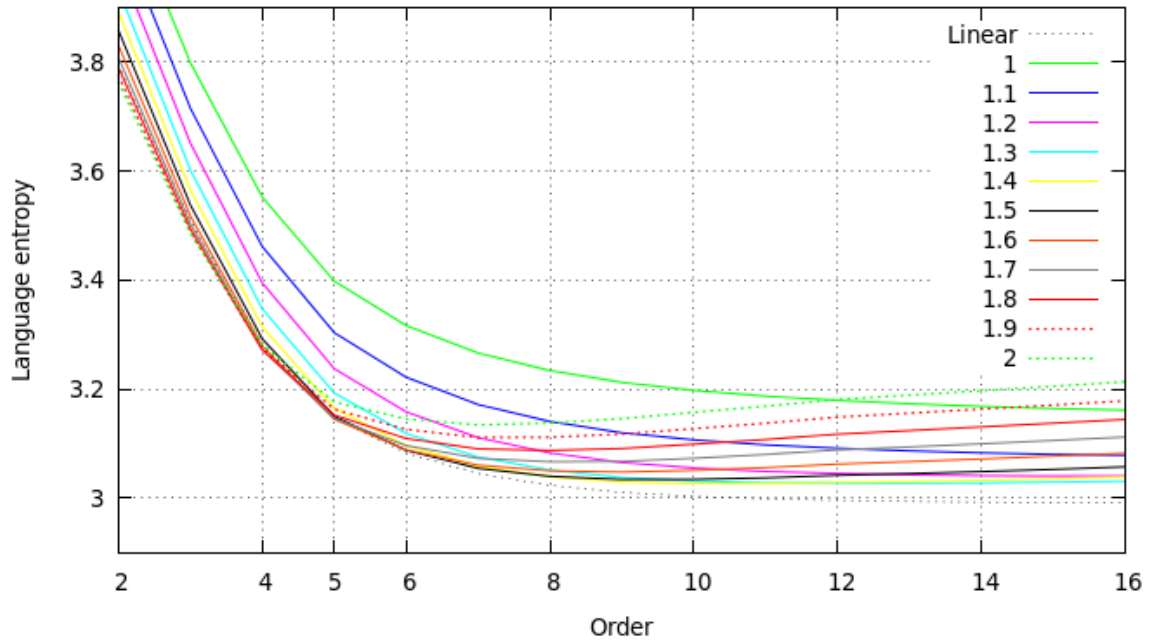


Figura 5.5: Entropía cruzada en función del orden del modelo, utilizando el método de mezcla lineal y exponencial con distintos valores de la constante r .

Lineal	$r = 1$	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	2
3,148	3,398	3,304	3,237	3,193	3,165	3,150	3,145	3,146	3,153	3,163	3,177

Cuadro 5.4: Entropía cruzada del modelo de orden 5 para la mezcla lineal, y la mezcla exponencial con distintos valores del parámetro r .

Sistema	Resultado
{seqint \otimes gis221}	1.91
{contour, seqint \otimes gis221}	1.94
{mode \otimes degree, pitch \otimes duration}	1.98
{pitch, seqint \otimes gis221}	1.99
{seqint \otimes gis221, mode \otimes degree}	2.02

Cuadro 5.5: Los cinco sistemas con menor entropía cruzada para el tipo **pitch**, usando la regla de combinación del producto.

5.2.2 Predicción de alturas

El primer experimento está limitado a la predicción del tipo **pitch**, es decir, los modelos deben predecir la secuencia de alturas de un ejemplo, conociendo su secuencia de duraciones. Los experimentos realizados por Conklin y Witten estaban limitados a la predicción del tipo **pitch**, luego, realizar este experimento permite comparar directamente los resultados respecto a aquellos de los autores originales.

Se excluyó de este experimento el punto de vista **duration**, dado que no aporta a la predicción del tipo **pitch**. Los puntos de vista enlazados tales como **pitch** \otimes **duration** sí son útiles a pesar de que sólo se esté prediciendo el tipo **pitch**, dado que las notas que se predicen tienen duraciones establecidas, de forma tal que este punto de vista permite predecir la altura en función de la duración.

El primer objetivo consistió en optimizar el modelo de largo plazo, sin utilizar el modelo de corto plazo. La tabla 5.5 muestra los cinco sistemas hallados que produjeron la menor entropía cruzada, empleando como método de combinación de puntos de vista la regla del producto. El valor mostrado es la entropía cruzada en bits obtenida para la predicción del tipo **pitch** en la colección de prueba.

Adicionalmente se probó la regla de combinación geométrica para combinar los puntos de vista. Este método tiene un parámetro b que se debe entonar explorando 11 valores posibles. En los experimentos realizados, se observó que este parámetro tiene un solo óptimo global, es decir, al encontrar un mínimo no es necesario seguir buscando. Esta observación se puede deducir también del artículo de Pearce *et al.* donde se introduce la regla de combinación geométrica [74].

Con cada valor probado de b , se evaluaron todos los conjuntos posibles de puntos de vista. La gráfica 5.6 muestra la menor entropía cruzada obtenida para cada valor de b .

La menor entropía cruzada fue de 1,62 y se obtuvo con $b = 1$. Es importante notar que para todos los valores de b probados la entropía cruzada fue considerablemente menor

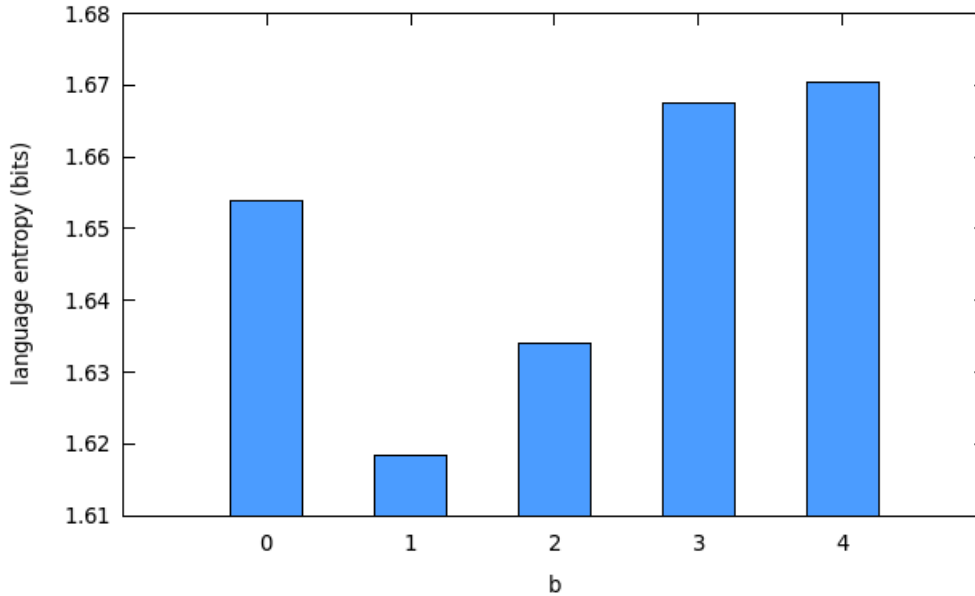


Figura 5.6: Optimización de la constante b para la combinación geométrica de puntos de vista, para predecir el tipo `pitch`.

que aquella obtenida con la regla del producto, lo cual indica que la regla geométrica es más efectiva.

Los cinco mejores sistemas de puntos de vista obtenidos con este valor de b se muestran en la tabla 5.6.

Otra observación importante se obtiene al comparar la entropía cruzada del sistema $\{\text{pitch} \otimes \text{duration}\}$, que se muestra en la tabla 5.7, con aquellas de la tabla 5.6. Este sistema consta de un solo punto de vista afín a la representación básica de los eventos, por lo tanto, resulta equivalente a usar un modelo de contexto sin el formalismo de los múltiples puntos de vista. Los resultados de la tabla 5.6 son superiores, lo cual demuestra que el formalismo de los sistemas de múltiples puntos de vista es efectivo.

Optimización del modelo de corto plazo

Se diseñó un nuevo experimento para, utilizando un modelo de largo plazo establecido, combinarlo con todos los modelos de corto plazo posibles, para así encontrar el mejor sistema general.

Para la combinación de modelos se utilizó la regla geométrica, y se buscó el valor óptimo de la constante b para este propósito. La figura 5.7 muestra la entropía mínima alcanzada con cada valor de b evaluado. El mejor resultado se obtuvo con $b = 8$, y en

Sistema	Resultado
$\{\text{seqint} \otimes \text{gis221}, \text{keysig} \otimes \text{pitch}, \text{mode} \otimes \text{seqint}, \text{mode} \otimes \text{degree}, \text{pitch} \otimes \text{duration}\}$	1.618
$\{\text{seqint} \otimes \text{gis221}, \text{keysig} \otimes \text{pitch}, \text{mode} \otimes \text{seqint}, \text{pitch} \otimes \text{duration}\}$	1.619
$\{\text{seqint}, \text{seqint} \otimes \text{gis221}, \text{keysig} \otimes \text{pitch}, \text{mode} \otimes \text{degree}, \text{pitch} \otimes \text{duration}\}$	1.620
$\{\text{seqint} \otimes \text{gis221}, \text{degree} \otimes \text{fib}, \text{keysig} \otimes \text{pitch}, \text{mode} \otimes \text{seqint}, \text{pitch} \otimes \text{duration}\}$	1.620
$\{\text{seqint} \otimes \text{gis221}, \text{degree} \otimes \text{seqint}, \text{keysig} \otimes \text{pitch}, \text{pitch} \otimes \text{duration}\}$	1.622

Cuadro 5.6: Los cinco sistemas con menor entropía cruzada para el tipo `pitch`, usando la regla de combinación geométrica con $b = 1$.

Sistema	Resultado
$\{\text{pitch} \otimes \text{duration}\}$	2.067

Cuadro 5.7: Entropía cruzada del sistema $\{\text{pitch} \otimes \text{duration}\}$ para el tipo `pitch`.

la tabla 5.8 se presentan los cinco mejores sistemas de corto plazo encontrados con este valor.

La menor entropía cruzada obtenida es de 1.52 bits/pitch. En consecuencia la incorporación del modelo de corto plazo produce una mejora en la capacidad predictiva.

Con fines comparativos, la tabla 5.9 muestra los resultados publicados por Conklin y Witten [27]. La entropía cruzada alcanzada por la presente implementación es 0,346 *bits/pitch* menor, en otras palabras 82 % menor. Es importante recordar que la entropía cruzada es una función logarítmica, por lo tanto incluso pequeñas variaciones en ella son significativas.

Sistema	Resultado
$\{\text{pitch}, \text{contour}, \text{degree}, \text{seqint} \otimes \text{gis221}, \text{degree} \otimes \text{fib}\}$	1.5243
$\{\text{contour}, \text{degree}, \text{seqint} \otimes \text{gis221}, \text{degree} \otimes \text{fib}, \text{keysig} \otimes \text{pitch}\}$	1.5243
$\{\text{pitch}, \text{contour}, \text{seqint} \otimes \text{gis221}, \text{degree} \otimes \text{fib}\}$	1.5245
$\{\text{contour}, \text{seqint} \otimes \text{gis221}, \text{degree} \otimes \text{fib}, \text{keysig} \otimes \text{pitch}\}$	1.5245
$\{\text{pitch}, \text{contour}, \text{seqint} \otimes \text{gis221}, \text{degree} \otimes \text{fib}, \text{mode} \otimes \text{degree}\}$	1.5258

Cuadro 5.8: Los cinco modelos de corto plazo con menor entropía cruzada para el tipo `pitch`, usando la regla geométrica con $b = 8$ para combinar los modelos.

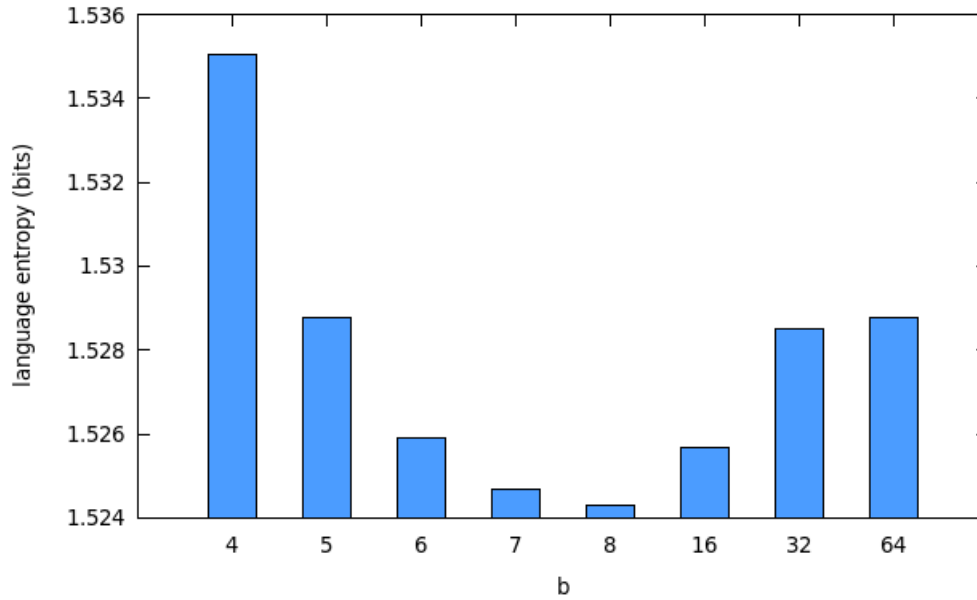


Figura 5.7: Optimización de la constante b para la combinación geométrica de los modelos de corto y largo plazo, para predecir el tipo `pitch`.

Sistema	Resultado
{ <code>pitch</code> }	2.05
{ <code>seqint</code> }	2.33
{ <code>seqint</code> \otimes <code>gis221</code> }	2.13
{ <code>seqint</code> \otimes <code>gis221</code> , <code>pitch</code> }	2.01
{ <code>intfref</code> \otimes <code>seqint</code> }	2.12
{ <code>intfref</code> \otimes <code>seqint</code> , <code>seqint</code> \otimes <code>gis221</code> }	1.94
{ <code>intfref</code> \otimes <code>seqint</code> , <code>seqint</code> \otimes <code>gis221</code> , <code>pitch</code> }	1.92
{ <code>intfref</code> \otimes <code>seqint</code> , <code>seqint</code> \otimes <code>gis221</code> , <code>pitch</code> , <code>intfref</code> \otimes <code>fib</code> }	1.87

Cuadro 5.9: Entropía cruzada del tipo `pitch` para diferentes sistemas de múltiples puntos de vista, publicada por Conklin y Witten [27].

Aunque en esencia el método usado aquí es igual al de Conklin y Witten, existen diferentes mejoras y ventajas con las cuales se contaba en este trabajo que permiten explicar la menor entropía cruzada obtenida. En primer lugar, la colección de melodías corales usada aquí es considerablemente más grande: 340 corales de entrenamiento y 10 de prueba versus 95 de entrenamiento y 5 de prueba usadas por Conklin y Witten.

Además, los archivos MusicXML utilizados incluían información sobre la tonalidad de cada pieza, mientras que Conklin y Witten sólo conocían la armadura de clave y no la tonalidad exacta. Esto permitió diseñar y utilizar nuevos puntos de vista basados en la tonalidad, como `degree`, `mode` \otimes `degree`, `mode` \otimes `seqint` y los tipos derivados de `degree`, los cuales contribuyeron a mejorar la capacidad predictiva.

Por otra parte, la técnica de combinación geométrica usada en este trabajo es más reciente que el trabajo original de Conklin y Witten. Además los autores originales utilizaron un modelo de orden 3 para el modelo de largo plazo y 2 para el modelo de corto plazo, mientras que aquí la mayor cantidad de memoria y velocidad de cómputo permitió usar modelos de tamaño mayor. Finalmente, en el presente trabajo se realizó una búsqueda exhaustiva de los conjuntos de puntos de vista, mientras que el trabajo original estaba enfocado en presentar el formalismo y no en hallar el conjunto de puntos de vista óptimo.

5.2.3 Predicción de alturas y duraciones

En el primer experimento se midió la capacidad de los modelos de predecir el tipo `pitch`. En este segundo experimento se mide la capacidad de predecir el tipo `pitch` \otimes `duration`, en otras palabras, los modelos deben predecir las notas completas.

Mientras que en el primer caso la cardinalidad del conjunto de eventos era $|\xi| = |\text{pitch}| = 20$, en este caso es igual a $|\xi| = |\text{pitch} \times \text{duration}| = 160$, lo cual aumenta significativamente la dificultad del problema.

Idealmente, un sistema interactivo debe ser capaz de generar alturas y duraciones, por lo tanto, encontrar en este experimento un modelo con buena capacidad predictiva es necesario para la premisa de este trabajo de crear un sistema de música interactiva.

Se utilizó la regla de combinación geométrica, y se buscó el valor óptimo de b hasta obtener el mejor resultado con $b = 2$. Los cinco mejores sistemas obtenidos se listan en la tabla 5.10. Todos ellos incluyen el tipo `duration`, así como los tipos `seqint` \otimes `gis221` y `pitch` \otimes `duration` que relacionan alturas con duraciones. Esto es consistente con el hecho de que los modelos ahora también deben predecir las duraciones.

La entropía cruzada del mejor modelo de largo plazo obtenido (cuadro 5.10) aparece

Sistema	Resultado
{duration, seqint \otimes gis221, degree \otimes seqint, keysig \otimes pitch, pitch \otimes duration}	2.962
{duration, seqint \otimes gis221, keysig \otimes pitch, mode \otimes seqint, pitch \otimes duration}	2.964
{duration, seqint \otimes gis221, keysig \otimes pitch, mode \otimes degree, pitch \otimes duration}	2.966
{duration, seqint \otimes gis221, keysig \otimes pitch, pitch \otimes duration}	2.967
{duration, seqint, seqint \otimes gis221, keysig \otimes pitch, pitch \otimes duration}	2.970

Cuadro 5.10: Los cinco sistemas con menor entropía cruzada promedio para el tipo pitch \otimes duration, utilizando combinación geométrica con $b = 2$.

Sistema	Resultado
{pitch, duration, seqint \otimes gis221, keysig \otimes pitch, pitch \otimes duration}	2.724
{pitch, duration, seqint \otimes gis221, degree \otimes seqint, pitch \otimes duration}	2.725
{duration, seqint \otimes gis221, degree \otimes seqint, keysig \otimes pitch, pitch \otimes duration}	2.725
{pitch, duration, seqint \otimes gis221, mode \otimes degree, pitch \otimes duration}	2.727
{duration, seqint \otimes gis221, keysig \otimes pitch, mode \otimes degree, pitch \otimes duration}	2.727

Cuadro 5.11: Los cinco modelos de corto plazo con menor entropía cruzada para el tipo pitch \otimes duration, utilizando la regla geométrica con $b = 1$ para combinar los modelos.

en letra negrita.

Optimización del modelo de corto plazo

Al igual que para la predicción del tipo pitch, se realizó un segundo experimento para optimizar el modelo de corto plazo. Se utilizó la regla de combinación geométrica para combinar los modelos. El valor óptimo de b obtenido en este caso fue $b = 1$. El cuadro 5.11 resume los 5 mejores sistemas obtenidos con esta configuración. El mejor resultado aparece en letra negrita.

5.3 Perfil entrópico

El perfil entrópico representa el valor de la entropía o sorpresa al predecir cada uno de los eventos de una secuencia preexistente. Si el modelo le asigna una alta probabilidad a un evento de la secuencia dado un contexto, la entropía para este evento será muy baja, en otras palabras la predicción del modelo habrá sido acertada. Si al contrario el modelo le asigna una baja probabilidad, la entropía será muy alta, en otras palabras el modelo habrá obtenido un alto nivel de sorpresa al encontrar este evento.

En consecuencia, la gráfica del perfil entrópico permite representar que tan predecible o impredecible fue cada uno de los eventos de una secuencia para el modelo.

La parte superior de la figura 5.8 muestra el perfil entrópico del coral BWV 284. La predicción se realizó utilizando el mejor modelo obtenido en el experimento anterior (el modelo resaltado en la tabla 5.11). Los primeros 25 eventos, es decir, la mitad del coral, fueron pasados al modelo de corto plazo como contexto. El modelo predijo los eventos 25 al 50. La parte inferior de la figura 5.8 muestra las notas del coral BWV 284 correspondientes a este segmento.

En la gráfica se puede observar que los eventos con mayor entropía fueron el 38 y el 47. El evento 38 es un *mib* con duración de corchea, y es primera vez que un *mib* y que una corchea aparecen en la pieza. Esto explica que sea difícil de predecir, además de que el re anterior hubiese podido resolver a un *do* que es la tónica.

El evento 47 es un *sol* antecedido de un *sib*. Se explica que este evento sorprenda al modelo dado que en todas las apariciones anteriores el *sib* era seguido de un *do* o un *la*. Por otra parte, como se vio en la sección 5.1, en las melodías corales predomina el movimiento por grado conjunto, por lo tanto el salto de una tercera menor descendente no es lo más frecuente. Además, desde el comienzo de ese compás las notas venían bajando por grado conjunto, y el *la* hubiese continuado con ese patrón.

5.4 Composición de melodías

El mejor modelo conseguido en la fase de optimización (el modelo resaltado en la tabla 5.11) fue usado para generar melodías nuevas. La generación es un problema más difícil que la predicción, dado que en la predicción el modelo sólo debe determinar una nota por cada ejemplo. En la generación, en cambio, se debe determinar una secuencia de notas. Esto significa que si el modelo genera una nota errónea, es decir, que no esté en correspondencia con la verdadera distribución de probabilidad, este error afectará la

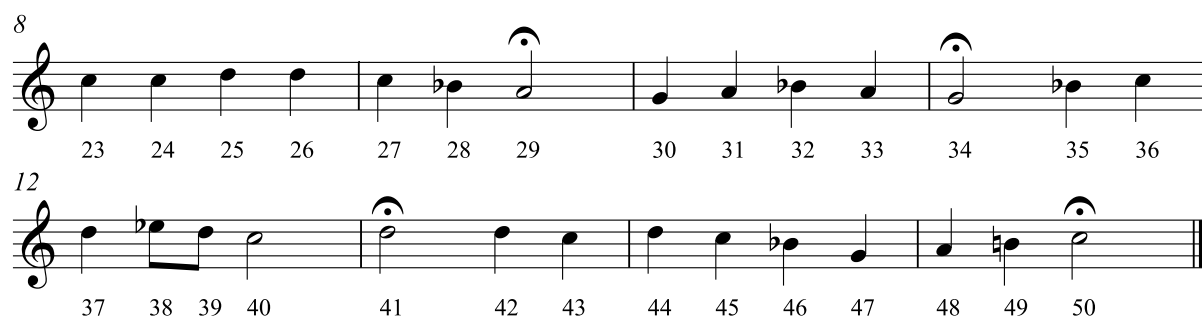
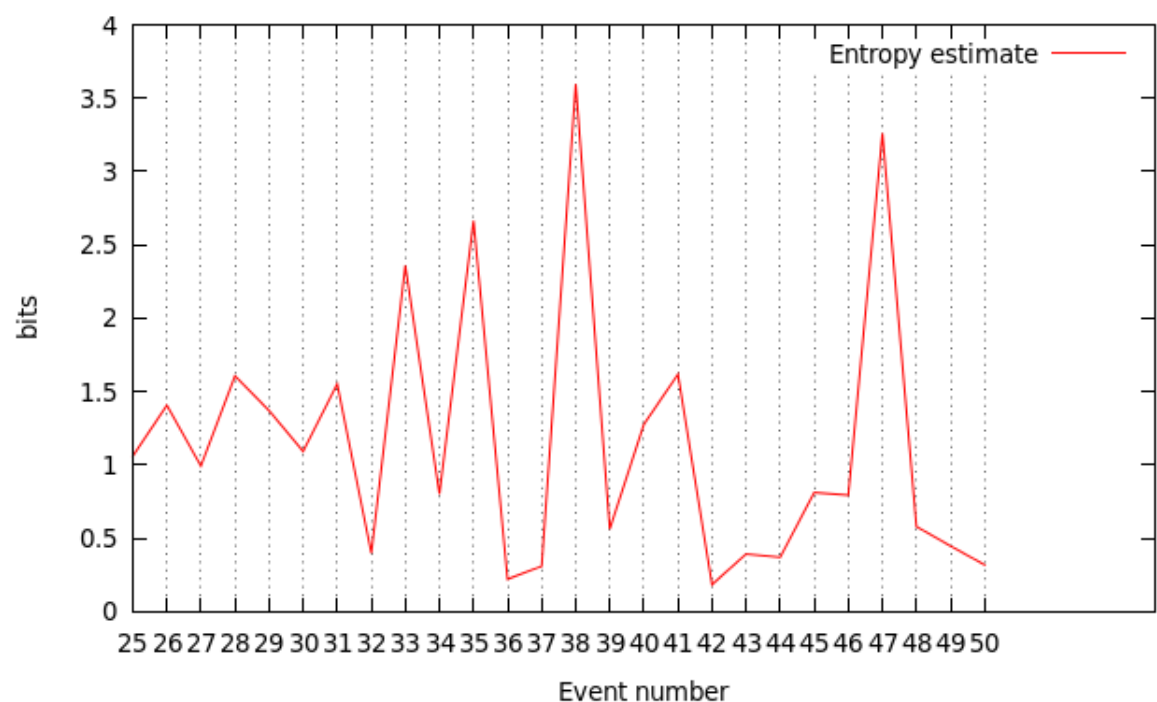


Figura 5.8: Perfil entrópico para el coral BWV 284 (arriba) y fragmento correspondiente del coral (abajo)

generación de la siguiente nota, y así sucesivamente.

En realidad, esto no es un problema de la generación por sí misma, sino que es una consecuencia del algoritmo de generación específico que está siendo utilizado de camino aleatorio, que genera las notas una por una de izquierda a derecha. Esta estrategia es voraz en el sentido de que maximiza la probabilidad de los eventos individuales en detrimento de la probabilidad de la secuencia, por lo tanto, pueden surgir este tipo de problemas.

Debido a esto, fue necesario ajustar el modelo predictivo obtenido en dos aspectos, para adecuarlo a la generación. El primer cambio realizado fue en el método de combinación de puntos de vista. Mientras que la regla de combinación geométrica proporciona buenos resultados para la predicción, para la generación resulta en secuencias inaceptables, por ejemplo, con notas audiblemente fuera de la escala. El método de combinación de puntos de vista fue cambiado a la regla del producto. Este operador es conjuntivo, en el sentido de que multiplica directamente las probabilidades obtenidas de cada modelo, por lo tanto, sólo los eventos con alta probabilidad según todos los modelos tienen posibilidad de ser seleccionados. Esto es una ventaja para la generación, donde no se pueden tolerar errores. Realizar este cambio resultó en una mejora significativa de la calidad de las melodías generadas.

El segundo cambio realizado fue en el método de combinación de modelos. Si bien de nuevo la regla geométrica es adecuada a este propósito en el caso de la predicción, para la generación el problema es que le confiere demasiada importancia al modelo de corto plazo. El resultado es que se produce un ciclo en el cual las notas que han sido generadas vuelven a ser generadas, y resultan melodías con una nota repetida *ad infinitum* o con un motivo corto que se repite incesantemente.

En este caso la solución fue sustituir la regla de combinación de modelos por una combinación lineal, que le asigna un peso de 0,8 al modelo de largo plazo y 0,2 al modelo de corto plazo. De esta manera las melodías tienen variedad, al mismo tiempo que el modelo de corto plazo les confiere consistencia interna.

El sistema descrito fue utilizado para generar 500 melodías de longitud $N = 50$. Cada melodía fue generada tomando como semilla una melodía de la colección de melodías corales. La primera frase (las notas hasta el primer calderón inclusive), la tonalidad y el compás (3/4 ó 4/4) de esta melodía eran usadas en la melodía generada. También, la primera frase era pasada como contexto al modelo de corto plazo.

De estas melodías, fueron elegidas cinco aleatoriamente para hacer una evaluación humana de las melodías. Se diseñó una encuesta en línea en la cual los jurados debían

	H1	C1	H2	C2	H3	C3	H4	C4	H5	C5
\bar{X}	6,33	5,57	7,00	6,14	8,00	4,62	6,14	5,52	5,81	6,33
s	1,59	2,16	2,10	1,90	1,55	2,09	2,20	2,04	2,09	1,96
marg. error ($\alpha = 5\%$)	0,72	0,98	0,95	0,87	0,71	0,95	1,00	0,93	0,95	0,89
$t(20)$	1,38		2,12		6,35		1,35		-1,67	
valor p	0,182		0,047		0,000		0,194		0,110	

Cuadro 5.12: Media, desviación estándar, margen de error al 95 % de confianza, valor t y valor p asociado de los puntajes. Las melodías H2 y H3 obtuvieron una ventaja estadísticamente significativa sobre sus contrapartes.

escuchar estas cinco melodías además de la cinco melodías originales, y asignarles un puntaje del 1 al 10.

La encuesta fue contestada por 21 personas con formación musical, en su mayoría estudiantes del Conservatorio Nacional de Música Juan José Landaeta. La mayoría de los encuestados había al menos culminado los estudios de Teoría y Solfeo, y algunos tenían más de 5 años de formación clásica en un instrumento. La edad promedio de los encuestados fue de 23,14 años, con una desviación estándar de 3,03.

La encuesta, que puede ser consultada en línea [38], estaba organizada en cinco páginas. En cada página se podía escuchar la melodía original, la melodía computarizada y la primera frase (que era común a ambas). Los usuarios no sabían cuál era la melodía original, de hecho, la mayoría no sabía siquiera que existían en la encuesta melodías generadas por una computadora.

Se le pidió a los usuarios que evaluaran la calidad melódica general de cada melodía, así como la coherencia de la primera frase con el resto de la melodía. Todos los jurados debían evaluar las diez melodías, asignándoles un puntaje del 1 al 10, donde 1 significaba muy pobre y 10 muy bueno.

El cuadro 5.12 muestra el puntaje promedio obtenido por cada una de las melodías humanas (H) y computarizadas (C), así como la desviación estándar. Se muestra también el margen de error para un nivel de confianza del 95 %. Esta información se representa gráficamente en la figura 5.9.

Se realizaron pruebas t de muestras dependientes [99] para cada par de melodías con el fin de determinar si existía una ventaja a favor de alguna de las melodías. Para las melodías 1, 4 y 5, no se encontró una diferencia significativa, dentro de un nivel de confianza del 95 %.

El análisis para la melodía 2 arrojó un valor t significativo ($t_{(20)} = 2,12, p < 0,05$), e igualmente para la melodía 3 ($t_{(20)} = 2,12, p < 0,001$). En ambos casos la melodía humana

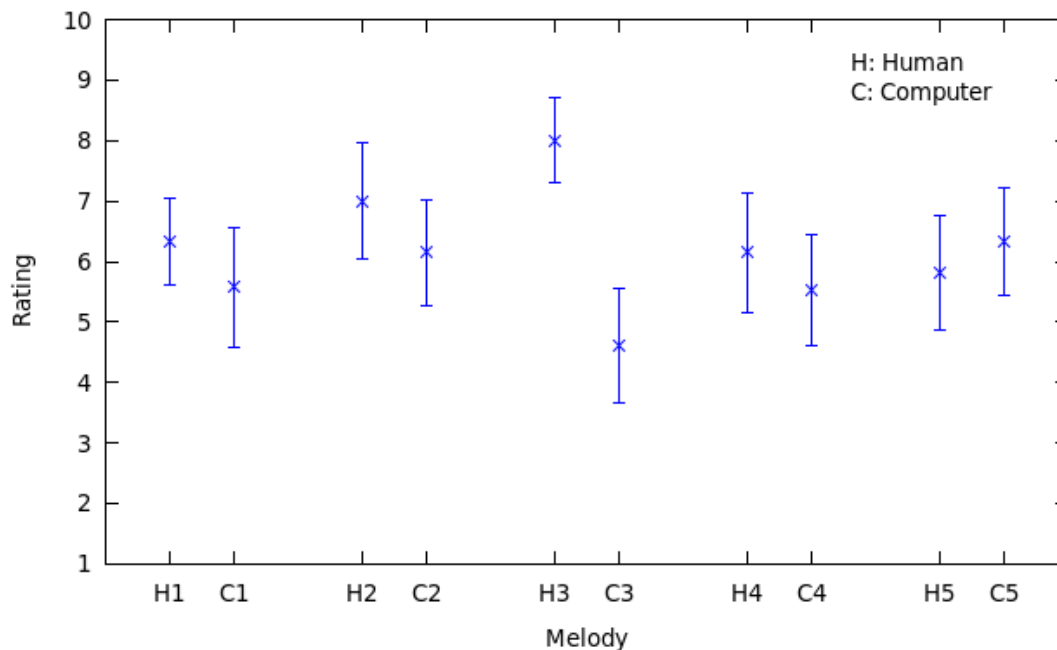


Figura 5.9: Evaluación humana de las melodías. Comparación entre las melodías humanas (H) y las melodías generadas por la computadora (C).

obtuvo un mayor puntaje promedio.

Desde el punto de vista descriptivo, salvo en el caso de la melodía 3, la diferencia entre el puntaje promedio de la melodía humana y computarizada siempre fue menor a 1 punto. Incluso sucedió que en el caso 5 la melodía computarizada obtuvo un mayor promedio.

Este resultado se considera satisfactorio, dado que existen distintas áreas en las cuales es posible mejorar el algoritmo de generación, por lo tanto se espera que en el futuro se pueda reducir la brecha entre las melodías computarizadas y humanas. Entre estas posibles mejoras está utilizar un algoritmo de generación no voraz y definir nuevos puntos de vista. También sería posible incorporar la variación de la entropía como un elemento del discurso musical, incluyéndola dentro de los modelos o dentro del algoritmo de generación. Así sería posible manipular el interés melódico y el grado de creatividad de las melodías.

La melodía computarizada que obtuvo el peor puntaje se muestra en la figura 5.10, junto con la melodía original. En la melodía computarizada, después del primer calderón, el motivo resaltado en rojo es interesante y consistente con la primera frase, sin embargo, lo correcto hubiese sido que el do y el re tuviesen duración de corchea, para imitar la primera frase. De esta manera la sensación rítmica de la melodía original no se hubiese perdido. El principal problema con la melodía generada es la repetición cuatro veces

consecutivas del motivo resaltado en azul. Esto hace perder el interés y evidentemente este tipo de repetición no es una práctica dentro del estilo. El motivo que comienza en el segundo tiempo del compás 12 ofrece contraste con el material anterior, por lo tanto es bien recibido. Sin embargo, lo correcto hubiese sido empezar en el cuarto tiempo y no en el segundo, para ser consistente con el carácter anacrúsico de la melodía.

El otro mayor problema con esta melodía es la ausencia de un sentido cadencial. La melodía original tiene una estructura de frases muy clara, donde cada frase termina en una cadencia, que se produce sobre distintos grados lo cual genera variedad. La melodía computarizada carece casi totalmente de esta sensación de reposo, excepto por el la ligado entre los compases 11 y 12, que en todo caso hubiese debido comenzar en cualquier tiempo menos el cuarto, para evitar la ligadura.

La figura 5.11 muestra la melodía computarizada que obtuvo el mayor puntaje, junto con la melodía original. Aunque en el experimento actual no se generaron calderones, la melodía generada tiene varios puntos que sirven de reposo, que se marcaron en rojo. Por lo tanto, esta melodía tiene mucho mayor sentido cadencial que la anterior. Un procedimiento muy efectivo en esta melodía es la repetición del motivo subrayado en azul, que al repetirse cambia la duración del do y el si a corchea, produciendo una variación que seguidamente resuelve por grado conjunto a una cadencia en sol. La reducción rítmica ayuda a crear una mayor expectativa hacia el sol.

El motivo que comienza al final del compás 10 tiene la misma cabeza que aquel que

Melodía coral BWV 394

Generación sobre la melodía coral BWV 394

Figura 5.10: Melodía 3 humana (arriba) y computarizada (abajo)

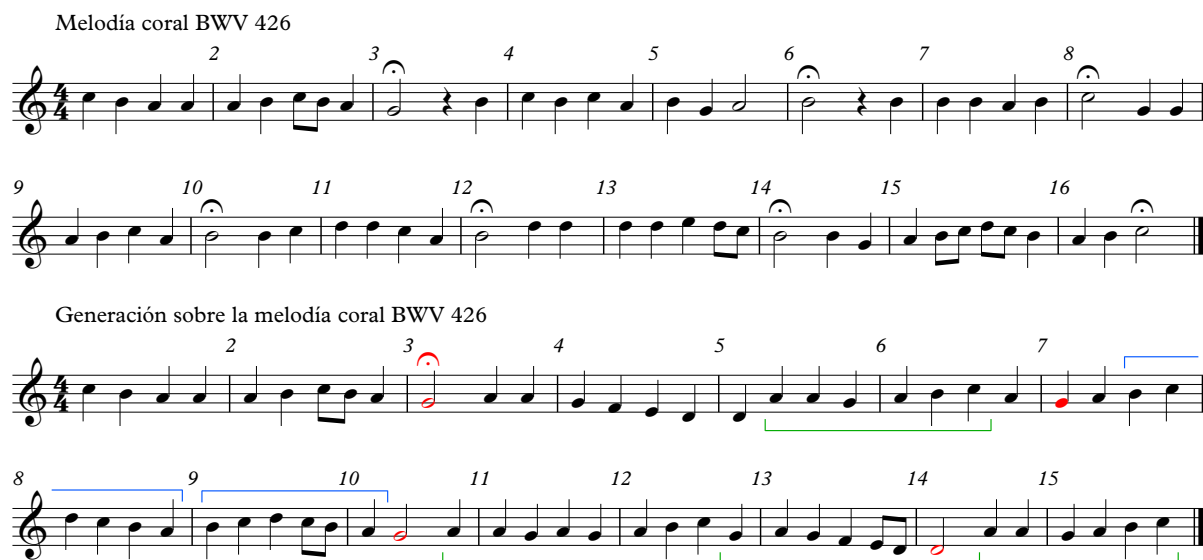


Figura 5.11: Melodía 5 humana (arriba) y computarizada (abajo)

comienza en el segundo tiempo del compás 5 (ambos subrayados en verde), y en general este motivo mantiene una muy buena coherencia con el resto de la melodía. El motivo del compás 5 se repite de nuevo en las últimas cinco notas de la melodía.

En general esta melodía tiene una buena consistencia interna y un sentido cadencial aceptable. Una desventaja es el exceso de movimiento por grado conjunto: la melodía original tiene ocho saltos (intervalos distintos de una segunda), mientras que la melodía computarizada sólo cuatro.

Con el fin de mostrar la capacidad del sistema de generalizar, en la dirección <http://ldc.usb.ve/~cgomez/bastien> fueron publicadas las 500 melodías generadas por el sistema.

5.5 Generación interactiva

La generación interactiva combina el modelo de largo plazo con el modelo interactivo. El usuario puede controlar el modelo interactivo mediante una interfaz de usuario sencilla basada en sliders, tal como fue descrito en la sección 4.6. En la generación interactiva no se utiliza el modelo de corto plazo, dado que el propósito es que el sistema responda rápidamente a las acciones del usuario. El modelo de corto plazo tendría el efecto de favorecer a los eventos que han sido generados anteriormente, lo cual reduciría la capacidad del sistema de responder rápidamente a las acciones del usuario.

Al inicio, todos los sliders están en la posición neutra. Esto significa que el sistema improvisa de acuerdo con las probabilidades del modelo de largo plazo únicamente. La versión actual del sistema tiene cuatro puntos de vista interactivos: **pitch**, **seqint**, **intfref** y **duration**. El desplazamiento hacia la derecha del slider de **pitch** aumenta proporcionalmente la altura de las notas generadas. Este efecto es aproximado: para una posición del slider de **pitch**, existe un rango relativamente amplio de alturas que pueden sonar. Por ejemplo, si el slider está en la posición más a la derecha, las notas inferiores del registro admitido por el sistema no son generadas.

El registro puede ser reforzado mediante el punto de vista **seqint**. Por ejemplo, si el slider de **pitch** está totalmente a la derecha, aumentar los intervalos contribuye a que las notas sean más altas. De nuevo, existe un grado de flexibilidad en la respuesta del sistema. Por ejemplo, si el slider de **seqint** está totalmente a la derecha, aún intervalos descendentes pueden ser generados, sobre todo si las notas generadas están en lo más alto del registro y no es posible seguir subiendo. Una desventaja del sistema es que el registro admitido de una octava más una quinta por encima del do central se siente muy limitado para la generación interactiva.

El punto de vista **duration** es uno de los más interesantes: en la posición inicial, el sistema genera mayormente negras. Al desplazar el slider de las duraciones hacia la izquierda, la frecuencia de las corcheas aumenta perceptiblemente. En el segmento izquierdo, este slider está bastante bien graduado: mientras más se desplaza hacia la derecha mayor es la frecuencia de las corcheas.

Lamentablemente lo mismo no ocurre en el sentido opuesto: incluso si el slider está totalmente hacia la derecha, escasamente el sistema genera notas blancas. Esto se debe a que en el modelo de largo plazo la probabilidad de las corcheas es mucho mayor que aquella de las blancas (estas últimas no son muy frecuentes en las melodías corales), por lo tanto, la contribución del sistema interactivo en este caso no es suficiente para hacer que se generen las blancas. Para resolver esto, sería necesario diseñar un nuevo método de combinación de modelos que en lugar de meramente multiplicar las probabilidades, garantice que la distribución de probabilidad final de los eventos siga más de cerca aquella del modelo interactivo. Es necesario realizar más investigación sobre métodos de combinación a ser aplicados para el modelo interactivo.

El punto de vista **intfref** también funciona adecuadamente: al desplazarlo hacia la izquierda las notas más extrañas a la escala suenan más frecuentemente (ver sección 4.7.1), mientras que al desplazarlo hacia la derecha la tónica y otras notas más propias de la escala

tienen mayor probabilidad de sonar. Este es uno de los puntos de vista interactivos más interesantes, dado que las notas extrañas a la escala tienen interés melódico y producen tensión. De los cuatro puntos de vista interactivos implementados, este es el más cercano al objetivo de poder manipular la emoción y el significado musical, sin embargo, todavía es necesario mucho trabajo futuro para lograr este propósito.

En general, el sistema interactivo es satisfactorio como una prueba de concepto. Una ventaja muy positiva es que siempre mantiene una improvisación musicalmente “correcta” independientemente de las acciones del usuario, sin errores obvios como notas disonantes en sucesión. Esto lo hace al mismo tiempo que responde perceptiblemente a las acciones del usuario. Por otra parte, el fin último de lograr una improvisación donde el usuario controle el carácter, el significado y las emociones en la música no se logrará sin antes aumentar la capacidad del sistema de expresión musical. Esto incluye el manejo de la polifonía, la capacidad de modelar el interés melódico e incluso el tratamiento de distintos timbres (instrumentos) y de las dinámicas.

Capítulo 6

Conclusiones y recomendaciones

En este trabajo se resolvió el problema de optimizar sistemas de múltiples puntos de vista para modelar el estilo de una colección musical. Se logró minimizar la entropía cruzada del tipo `pitch` de una colección de melodías corales a 1,52 *bits/pitch*, un resultado mejor que los 1,87 *bits/pitch* obtenidos por Conklin y Witten [27], Hall [41] y Cherla *et al.* [13]. Distintas razones contribuyeron a esta mejora: el recorrido exhaustivo del espacio de búsqueda de los modelos, la incorporación de la regla de combinación geométrica y la disponibilidad de un mayor número de datos de ejemplo.

Se encontraron casos en los cuales la estrategia de reducir la entropía cruzada no producía mejores resultados desde el punto de vista musical al momento de generar melodías. Esto es consecuencia de que la generación es más difícil que la predicción. Existen dos vertientes hacia la solución de este problema. La primera es utilizar otros algoritmos de generación, como VNS (*variable neighborhood search*), en lugar del camino aleatorio. El problema con el camino aleatorio es que es un algoritmo voraz que maximiza la probabilidad de los eventos individuales en lugar de aquella de la secuencia completa.

La otra vertiente consiste en diseñar nuevas medidas que permitan evaluar la calidad de un modelo predictivo. En este sentido, un buen modelo predictivo debe tener buena precisión y *recall*. El *recall* significa que el modelo debe asignarle una alta probabilidad a los eventos más probables en la verdadera distribución de probabilidad. La precisión significa que el modelo debe asignarle una baja probabilidad a los eventos improbables en la verdadera distribución. El problema con el uso de la entropía cruzada como función objetivo es que ella es fundamentalmente una medida del *recall*, y sólo indirectamente de la precisión. Por lo tanto es necesario desarrollar nuevas medidas que posean un buen balance entre precisión y *recall*.

La mayor capacidad predictiva alcanzada permitió aplicar los modelos no sólo para predecir alturas, sino para generar notas completas, es decir, para componer melodías. Esto supera los resultados obtenidos en el trabajo original, en el cual sólo se predecían alturas, y es una prueba de que es posible utilizar el formalismo de los múltiples puntos de vista como un algoritmo de composición automática.

Las melodías generadas fueron sometidas a una evaluación humana. De las cinco melodías juzgadas, en tres casos no se encontró una diferencia estadísticamente significativa entre el puntaje de la melodía generada y la melodía humana. Se analizaron los puntos fuertes y las deficiencias de las melodías generadas, y esto puede aplicarse como una estrategia para diseñar nuevos puntos de vista que permitan superar las deficiencias encontradas.

Los modelos generados fueron utilizados para la improvisación musical en tiempo real. Se extendió la teoría de los múltiples puntos de vista al caso interactivo, mediante un formalismo que puede ser usado en diversas aplicaciones. El esquema desarrollado puede ser aplicado con diversos tipos de interfaz de usuario, que no están limitados a interfaces tradicionales y que podrían aprovechar la interacción multimodal.

La principal limitación encontrada del formalismo de los múltiples puntos de vista es su postulado de modelar un estilo general y un estilo específico global para la pieza. La música suele tener una estructura jerárquica, formada por células, motivos, frases y temas, en otras palabras, en un discurso musical no existe una sola tendencia a lo largo de toda la pieza, sino que las estructuras musicales suelen tener carácter local. Dicho de otra manera, hace falta un análisis de mayor granularidad para poder representar fielmente la estructura de una pieza musical. Es posible concebir que en el futuro la noción de modelo de corto plazo se extienda a aquella de varios modelos locales, para así mejorar esta situación.

Se ilustró cómo el programa desarrollado y el formalismo de los múltiples puntos de vista pueden ser una herramienta útil para el analista musical. Definiendo y visualizando nuevos puntos de vista, sería posible revelar nuevas propiedades acerca de una obra musical. Esto permitiría verificar objetivamente postulados acerca de la música de práctica común que han sido conocidos por siglos, y lo que es más interesante, permitiría estudiar rasgos estilísticos de la música contemporánea, sobre la cual suelen existir menos estudios.

6.1 Trabajo futuro

El objetivo a largo plazo de este trabajo es desarrollar sistemas de música interactiva que sean una herramienta de colaboración creativa para los usuarios. Posiblemente, el primer paso en esta dirección sea sustituir los modelos de contexto por el método de predicción basado en RBM (*restricted Boltzmann machine*) propuesto por Cherla *et al.* [13]. Los autores de este trabajo demostraron que este método escala mejor, además de que tiene menos requerimientos de memoria y tiempo de cómputo. Todavía es necesario comprobar si su método puede ser usado eficazmente para predecir alturas y duraciones, dado que sólo ha sido aplicado para predecir el tipo `pitch`.

El siguiente paso sería incorporar la predicción de música polifónica. Whorley *et al.* [100] midieron la complejidad de armonizar melodías a cuatro voces, y realizaron algunos experimentos de generación. La complejidad computacional es un mayor problema en este caso, por lo tanto, aplicar el método de predicción basado en RBM, que escala mejor en función del tamaño del alfabeto, sería una gran ventaja.

Por último, sería necesario mejorar los métodos de predicción interactiva. Actualmente, el modelo interactivo determina la probabilidad de un evento sin tomar en cuenta el contexto, es decir, realiza una predicción de orden cero. Aplicando un algoritmo de agrupamiento automático, sería posible clasificar automáticamente de manera no supervisada las secuencias presentes en la colección musical, y en base a esto obtener clases que formarían el dominio del sistema interactivo. El modelo interactivo sería más efectivo si además se toma en cuenta el significado o carácter musical asociado a cada clase.

De acuerdo con Meyer, el significado musical se origina por la negación de expectativas gestadas en la pieza. Como ya fue señalado por Conklin y Witten, y por Wiggins [101], una pieza que siempre sea altamente predecible o mantenga una baja entropía produciría un discurso musical pobre. Por lo tanto, es necesario también idear nuevos algoritmos de generación que modelen la variación de la entropía a lo largo de una pieza.

En el actual trabajo sólo se incluyó una discusión sobre el desempeño del modelo interactivo, pero en el futuro es necesario realizar pruebas con usuarios que juzguen la calidad de este sistema: su grado de creatividad, su capacidad de responder correctamente a las acciones del usuario, la calidad, variedad e interés de la música generada y la capacidad del sistema de responder a emociones, caracteres o estados de ánimo.

Se concluye con una visión sobre la trayectoria futura de los métodos de composición automática. La figura 6.1 muestra distintos niveles de significado musical. El primero es el nivel sintáctico: a este nivel una pieza es válida si cumple con ciertas reglas básicas de la

notación musical. El siguiente es el nivel del lenguaje musical, que en el caso de la música tonal incluye la armonía, la construcción melódica, el contrapunto, las formas musicales y en general las reglas universales y bien establecidas sobre la música tonal.

La mayoría de los métodos de composición automática desarrollados hasta el presente llegan hasta este segundo peldaño. Como ha sido discutido anteriormente en este texto, la música tiene un significado incorporado, que de acuerdo con Meyer está íntimamente ligado con aspectos psicológicos de la percepción musical, como la creación de expectativas y su posterior satisfacción o inhibición. Ya ha sido discutido en este texto que la entropía pudiera ayudar a modelar este aspecto del discurso musical.

De acuerdo con Meyer, la música tiene también un significado denotativo, que hace referencia a elementos extramusicales como conceptos, acciones, estados emocionales y carácter [61]. Existen pocos trabajos sobre modelado computacional de las emociones en la música (ver [34] para un ejemplo), y la mayoría han sido realizados con audio y no con música simbólica. Sin embargo, no hay duda de que las emociones y otros tipos de significado denotativo son susceptibles a ser modelados computacionalmente.

Muchos alegan que un problema con este tipo de significados es que varían según el oyente y son determinados por sus experiencias, pero Meyer contesta que el significado puramente musical también depende de la experiencia y de la cultura del oyente.

¿Tiene el arte un significado universal? Esta ha sido una pregunta central en las teorías estéticas de los últimos dos siglos. Un concepto central del romanticismo es el mito del alma, según el cual existen formas universales e innatas en el fondo de todas las personas [9]. El arte contemporáneo, en cambio, exalta la individualidad del artista, y la importancia yace en la forma del objeto de arte en sí, desligado de cualquier significado.

Existen dos enfoques para el diseño de un sistema de música interactiva: el primero está centrado en la comunidad de usuarios en general, de forma tal que estos significados musicales y el mapeo entre música e interacción sean fijos y comunes a todos los usuarios. El segundo enfoque consiste en ofrecerle al usuario una experiencia personalizada, donde el sistema puede ser configurado o se adapta a los gustos de cada usuario.

En el segundo caso, el sistema se adaptaría más a la estética particular de cada usuario, y las composiciones resultarían más creativas y variadas. El análisis automático de las preferencias y composiciones entre usuarios, y la búsqueda de similitudes entre ellas, podría darnos una noción objetiva de qué tan universal es el lenguaje musical.

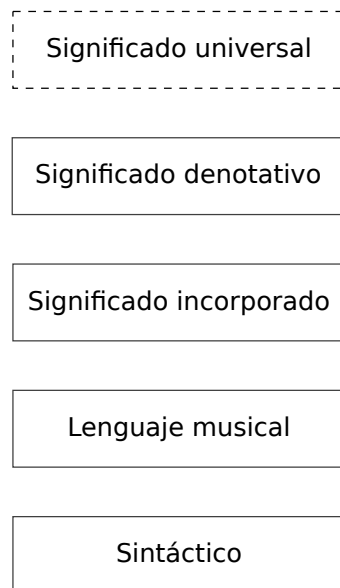


Figura 6.1: Distintos niveles de significado de la música

Bibliografía

- [1] C. Ames, “Automated composition in retrospect: 1956-1986,” *Leonardo*, vol. 20, no. 2, pp. 169–185, 1987.
- [2] —, “The markov process as a compositional model: A survey and tutorial,” *Leonardo*, vol. 22, no. 2, pp. 175–187, 1989.
- [3] C. Anagnostopoulou and G. Westermann, “Classification in music: A computational model for paradigmatic analysis,” in *Proceedings of the International Computer Music Conference*, 1997, pp. 125–128.
- [4] G. Assayag and S. Dubnov, “Using factor oracles for machine improvisation,” *Soft Comput.*, vol. 8, pp. 604–610, September 2004.
- [5] G. Assayag, S. Dubnov, and O. Delerue, “Guessing the composer’s mind: Applying universal prediction to musical style,” in *Proceedings of the International Computer Music Conference*, 1999.
- [6] M. Astor. (2014, 3) Contrapunto para hoy. [Online]. Available: <http://ucv.academia.edu/MiguelAstor/>
- [7] M. Baroni, S. Maguire, and W. Drabkin, “The concept of musical grammar,” *Music Analysis*, vol. 2, no. 2, pp. 175–208, 1983.
- [8] R. Begleiter, R. El-Yaniv, and G. Yona, “On prediction using variable order Markov models,” *J. Artif. Intell. Res.*, vol. 22, pp. 385–421, 2004.
- [9] A. Béguin, *El alma romántica y el sueño. Ensayo sobre el romanticismo alemán y la poesía francesa*. F.C.E., 1981.
- [10] M. A. Boden, “Computer models of creativity,” *AI Magazine*, vol. 30, no. 3, pp. 23–34, 2009.
- [11] G. Boenn, M. Brain, M. D. Vos, and J. Fitch, “Automatic composition of melodic and harmonic music by answer set programming,” in *Proceedings of the 24th International Conference on Logic Programming, Udine, Italy*, 2008.
- [12] S. Bushinsky, “Deus ex machina - A higher creative species in the game of chess,” *AI Magazine*, vol. 30, no. 3, pp. 63–70, 2009.

- [13] S. Cherla, T. Weyde, A. d. Garcez, and M. Pearce, “A distributed model for multiple-viewpoint melodic prediction,” in *ISMIR*, 2013.
- [14] P. Chordia, A. Sastry, T. Mallikarjuna, and A. Albin, “Multiple viewpoints modeling of tabla sequences,” in *ISMIR*, 2010.
- [15] T. Ciufo, “Design concepts and control strategies for interactive improvisational music systems,” in *Proceedings of the MAXIS International Festival*, 2003.
- [16] J. G. Cleary and W. J. Teahan, “Unbounded length contexts for PPM,” *The Computer Journal*, vol. 40, no. 2 and 3, pp. 67–75, 1997.
- [17] H. Cohen, *Explorations in Art and Technology*. Springer, 2002, ch. A million millennial Medicis.
- [18] N. Collins, “Towards autonomous agents for live computer music: Realtime machine listening and interactive music systems,” Ph.D. dissertation, University of Cambridge, 2006.
- [19] S. Colton, “The HR program for theorem generation,” *Lecture Notes in Computer Science*, vol. 2392, pp. 285–289, 2002.
- [20] S. Colton, R. L. de Mántaras, and O. Stock, “Computational creativity: Coming of age,” *AI Magazine*, vol. 30, no. 3, pp. 11–14, 2009.
- [21] D. Conklin, “Music generation from statistical models,” in *Proceedings of the AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, 2003, p. 30–35.
- [22] —, “Representation and discovery of vertical patterns in music,” in *Proceedings of the Second International Conference on Music and Artificial Intelligence*. London: Springer-Verlag, 2002, pp. 32–42.
- [23] —, “Melodic analysis with segment classes,” *Mach. Learn.*, vol. 65, pp. 349–360, December 2006.
- [24] —, “Multiple viewpoint systems for music classification,” *Journal of New Music Research*, vol. 42, no. 1, pp. 19–26, 2013.
- [25] —. (2014, 3) Darrell Conklin home page. [Online]. Available: <http://www.ehu.es/cs-ikerbasque/conklin/>
- [26] D. Conklin and M. Bergeron, “Discovery of contrapuntal patterns,” in *ISMIR*, 2010.
- [27] D. Conklin and I. H. Witten, “Multiple Viewpoint Systems for Music Prediction,” *Journal of New Music Research*, vol. 24, no. 1, 1995.
- [28] N. Cook, *A Guide to Musical Analysis*. New York: George Braziller, 1987.

- [29] D. Cope, *Virtual Music: Computer Synthesis of Musical Style*. MIT Press, 2001.
- [30] R. Crandall and C. Pomerance, *Prime numbers: a computational perspective*. Springer, 2005.
- [31] L. Cuddy and C. Lunney, “Expectancies generated by melodic intervals: Perceptual judgments of melodic continuity,” *Attention, Perception, & Psychophysics*, vol. 57, pp. 451–462, 1995.
- [32] T. G. Dietterich, “Ensemble methods in machine learning,” in *Proceedings of the First International Workshop on Multiple Classifier Systems*, 2000, pp. 1–15.
- [33] W. Duch, “Intuition, insight, imagination and creativity,” *Computational Intelligence Magazine, IEEE*, vol. 2, no. 3, pp. 40–52, 2007.
- [34] T. Eerola, O. Lartillot, and P. Toivainen, “Prediction of multidimensional emotional ratings in music from audio using multivariate regression models,” in *ISMIR*, 2009, pp. 621–626.
- [35] E. Glenn and Schellenberg, “Expectancy in melody: Tests of the implication-realization model,” *Cognition*, vol. 58, no. 1, pp. 75 – 125, 1996.
- [36] G. Geiger, “Using the touch screen as a controller for portable computer music instruments,” in *Proceedings of the 2006 conference on New interfaces for musical expression*. Paris: IRCAM – Centre Pompidou, 2006, pp. 61–64.
- [37] P. Gervás, “Computational approaches to storytelling and creativity,” *AI Magazine*, vol. 30, no. 3, pp. 49–62, 2009.
- [38] C. Gómez. Evaluación de melodías. [Online]. Available: <https://docs.google.com/forms/d/1yWZS9mDR5S-CAwuXLdL1Z2D-AruF1Za62LKEXj3Jlfo/viewform>
- [39] M. Good, “MusicXML for notation and analysis,” *The virtual score: representation, retrieval, restoration*, vol. 12, pp. 113–124, 2001.
- [40] M. Greentree. (2014, 4) Bach chorales. [Online]. Available: <http://www.jsbchorales.net/>
- [41] M. A. Hall, “Selection of attributes for modeling Bach chorales by a genetic algorithm,” in *Proceedings of the Second International Conference on Artificial Neural Networks and Expert Systems*, 1995, pp. 182–185.
- [42] D. Herremans, K. Sörensen, and D. Conklin, “First species counterpoint music generation with VNS and vertical viewpoints,” in *ORBEL 28 (Belgian Association of Operations Research)*, 2014.

- [43] H. Hild, J. Feulner, and W. Menzel, “HARMONET: A neural net for harmonizing chorales in the style of J. S. Bach,” *Advances in Neural Information Processing Systems*, vol. 4, pp. 267–274, 1993.
- [44] L. M. Hiller, L. A.; Isaacson, “Musical composition with a high-speed digital computer,” *J. Audio Eng. Soc.*, vol. 6, no. 3, pp. 154–160, 1958.
- [45] D. R. Hofstadter, *Virtual Music: Computer Synthesis of Musical Style*. MIT Press, 2001, ch. Staring Emmy Straight in the Eye – And Doing My Best Not to Flinch.
- [46] —, *Gödel, Escher, Bach: Un Eterno y Grácil Bucle*. Tusquets, 2007.
- [47] P. Hudak, *The Haskell School of Music – From Signals to Symphonies*. Yale University, 2013.
- [48] P. Hudak, A. Courtney, H. Nilsson, and J. Peterson, “Arrows, robots, and functional reactive programming,” in *Advanced Functional Programming*. Springer, 2003, pp. 159–187.
- [49] J. Hughes, “Programming with arrows,” in *Advanced Functional Programming*. Springer, 2005, pp. 73–129.
- [50] G. Hutton, *Programming in Haskell*. Cambridge University Press, 2007.
- [51] L. G. Kraft, “A device for quantizing, grouping, and coding amplitude-modulated pulses,” Ph.D. dissertation, Massachusetts Institute of Technology, 1949.
- [52] P. Kugel, “Myhill’s thesis: There’s more than computing in musical thinking,” *Computer Music Journal*, vol. 14, no. 3, pp. 12–25, 1990.
- [53] O. Lartillot, S. Dubnov, G. Assayag, G. Bejerano, and B. Gurion, “Automatic modeling of musical style,” in *Proceedings of the International Computer Music Conference*, 2001.
- [54] F. Lerdahl, R. Jackendoff, and R. Jackendoff, *A generative theory of tonal music*. MIT Press, 1996.
- [55] H. Liu, E. Cheng, and P. Hudak, “Causal commutative arrows and their optimization,” in *ACM Sigplan Notices*, vol. 44, no. 9, 2009, pp. 35–46.
- [56] B. Z. Manaris, P. Roos, P. Machado, D. Krehbiel, L. Pellicoro, and J. Romero, “A corpus-based hybrid approach to music analysis and composition,” in *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, 2007.
- [57] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*. MIT press, 1999, ch. Mathematical foundations, p. 73.

- [58] A. Marsden, “Automatic derivation of musical structure: A tool for research on Schenkerian analysis,” in *Proceedings of the 8th International Conference on Music Information Retrieval*, 2007.
- [59] P. Mavromatis and M. Brown, “Parsing context-free grammars for music: A computational model of Schenkerian analysis,” in *Proceedings of the 8th International Conference on Music Perception & Cognition*, 2008.
- [60] B. McMillan, “Two inequalities implied by unique decipherability,” *IRE Transactions on Information Theory*, vol. 2, no. 4, pp. 115–116, 1956.
- [61] L. Meyer, *Emotion and meaning in music*. University of Chicago Press, 1961.
- [62] L. B. Meyer, “Meaning in music and information theory,” *The Journal of Aesthetics and Art Criticism*, vol. 15, no. 4, pp. 412–424, 1957.
- [63] E. R. Miranda and J. A. Biles, Eds., *Evolutionary Computer Music*. Springer, 2007.
- [64] E. Narmour, *The analysis and cognition of melodic complexity: The implication-realization model*. University of Chicago Press, 1992.
- [65] A. Newell, J. C. Shaw, and H. A. Simon, *Contemporary Approaches to Creative Thinking*. New York: Atherton, 1963, ch. The Process of Creative Thinking, pp. 63–119.
- [66] H. Newton-Dunn, H. Nakano, and J. Gibson, “Block Jam: A tangible interface for interactive music,” in *Proceedings of the 2003 conference on New interfaces for musical expression*, 2003, pp. 170–177.
- [67] G. Nierhaus, *Algorithmic composition: Paradigms of automated music generation*. Springer, 2009.
- [68] F. Pachet, “Beyond the cybernetic jam fantasy: The Continuator,” *IEEE Comput. Graph. Appl.*, vol. 24, pp. 31–35, January 2004.
- [69] —, “Interacting with a musical learning system: The Continuator,” *Lecture Notes in Computer Science*, vol. 2445, 2002.
- [70] F. Pachet and P. Roy, “Musical harmonization with constraints: A survey,” *Constraints*, vol. 6, no. 1, pp. 7–19, 2001.
- [71] —, “Markov constraints: Steerable generation of Markov sequences,” *Constraints*, vol. 16, pp. 148–172, 2011.
- [72] G. Papadopoulos and G. Wiggins, “AI methods for algorithmic composition: A survey, a critical view and future prospects,” in *Proc. AISB’99 Symp. Musical Creativity*, 1999, pp. 110–117.

- [73] D. Pazel, S. Abrams, and R. Fuhrer, “A distributed interactive music application using harmonic constraint,” in *Proceedings of the International Computer Music Conference*, 2000.
- [74] M. Pearce, D. Conklin, and G. Wiggins, “Methods for combining statistical models of music,” in *Computer Music Modeling and Retrieval*, ser. Lecture Notes in Computer Science. Springer, 2005, vol. 3310, pp. 295–312.
- [75] S. L. Peyton Jones and A. L. M. Santos, “A transformation-based optimiser for Haskell,” *Science of Computer Programming*, vol. 32, no. 1, pp. 3–47, 1998.
- [76] A. Pienimäki and K. Lemström, “Clustering symbolic music using paradigmatic and surface level analyses,” in *Proceedings of the 5th International Conference on Music Information Retrieval*, 2004.
- [77] J.-C. Pomerol and F. Adam, “Understanding human decision making – a fundamental step towards effective intelligent decision support,” in *Intelligent Decision Making: An AI-Based Approach*, G. Phillips-Wren, N. Ichalkaranje, and L. Jain, Eds. Springer, 2008, pp. 3–40.
- [78] D. Ponsford, G. Wiggins, and C. Mellish, “Statistical learning of harmonic movement,” *Journal of New Music Research*, vol. 28, pp. 150–177, 1999.
- [79] L. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, feb 1989.
- [80] G. Ritchie, “Can computers create humor?” *AI Magazine*, vol. 30, no. 3, pp. 71–81, 2009.
- [81] K. Robinson, *Out of our minds: Learning to be creative*. Capstone, 2011.
- [82] R. Rowe, “The aesthetics of interactive music systems,” *Contemporary Music Review*, vol. 18, no. 3, pp. 83–87, 1999.
- [83] N. Ruwet, “Méthodes d’analyse en musicologie,” *Revue belge de Musicologie / Belgisch Tijdschrift voor Muziekwetenschap*, pp. 65–90, 1966.
- [84] S. Sadie, Ed., *The New Grove Dictionary of Music and Musicians*. London: Macmillan, 1980.
- [85] W. Schulze and B. van der Merwe, “Music generation with Markov models,” *IEEE Multimedia*, vol. 18, pp. 78–85, 2011.
- [86] C. E. Shannon, “Prediction and entropy of printed English,” *Bell Syst. Tech. J.*, vol. 30, pp. 50–64, 1951.
- [87] D. Shkarin, “PPM: One step to practicality,” in *Proceedings of the IEEE Data Compression Conference*, 2002, pp. 202–211.

- [88] H. A. Simon, “Explaining the ineffable: AI on the topics of intuition, insight and inspiration,” in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1995, pp. 939–949.
- [89] M. Steedman, *The blues and the abstract truth: Music and mental models*. Erlbaum, 1996, ch. Mental Models in Cognitive Sciences.
- [90] M. J. Steedman, “A generative grammar for jazz chord sequences,” *Music Perception*, vol. 2, no. 1, pp. 52–77, 1984.
- [91] M. Straka, “The performance of the Haskell containers package,” in *ACM Sigplan Notices*, vol. 45, no. 11, 2010, pp. 13–24.
- [92] A. Tanaka, N. Tokui, and A. Momeni, “Facilitating collective musical creativity,” in *Proceedings of the 13th annual ACM international conference on Multimedia*, 2005, pp. 191–198.
- [93] D. M. Tax, M. Van Breukelen, R. P. Duin, and J. Kittler, “Combining multiple classifiers by averaging or by multiplying?” *Pattern recognition*, vol. 33, no. 9, 2000.
- [94] D. Temperley, *The cognition of basic musical structures*. MIT Press, 2004.
- [95] ———, *Music and Probability*. MIT Press, 2007.
- [96] S. Thrun. (2009) Google’s driverless car. [Online]. Available: http://www.ted.com/talks/sebastian_thrun_google_s_driverless_car.html
- [97] J. Triviño-Rodríguez and R. Morales-Bueno, “Using multiattribute prediction suffix graphs to predict and generate music,” *Computer Music Journal*, vol. 25, no. 3, pp. 62–79, 2001.
- [98] A. M. Turing, “Computing machinery and intelligence,” *Mind*, vol. 59, no. 236, pp. 433–460, 1950.
- [99] T. C. Urdan, *Statistics in plain English*. Taylor & Francis, 2005.
- [100] R. P. Whorley, G. A. Wiggins, C. Rhodes, and M. T. Pearce, “Multiple viewpoint systems: Time complexity and the construction of domains for complex musical viewpoints in the harmonization problem,” *Journal of New Music Research*, vol. 42, no. 3, pp. 237–266, 2013.
- [101] G. Wiggins, M. T. Pearce, and D. Müllensiefen, *The Oxford Handbook of Computer Music*. Oxford University Press, 2009, ch. Computational Modelling of Music Cognition and Musical Creativity, pp. 383–420.
- [102] T. Winkler, “Making motion musical: Gesture mapping strategies for interactive computer music,” in *Proceedings of the International Computer Music Conference*, 1995.

- [103] —, “Creating interactive dance with the Very Nervous System,” in *Proceedings of the Connecticut College Symposium on Arts and Technology*, 1997.
- [104] I. H. Witten and T. Bell, “The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression,” *IEEE Transactions on Information Theory*, vol. 37, no. 4, pp. 1085–1094, 1991.

Glosario de términos musicales

Este glosario proporciona una definición de algunos de los términos musicales utilizados en este informe.

acorde Combinación de notas tocadas simultáneamente.

acorde de tónica Acorde con el primer grado de la escala como raíz, formado por los grados $\hat{1}$, $\hat{3}$ y $\hat{5}$.

acorde mayor Tríada cuya tercera es mayor y quinta es justa.

anacrusa Nota o grupo de notas no acentuadas al comienzo de una frase de música.

calderón Símbolo musical cuyo efecto es “detener” momentáneamente el tempo. Las notas bajo un calderón se mantienen por un tiempo generalmente mayor a su valor escrito, arbitrariamente largo de acuerdo con la libertad del intérprete.

escala Sucesión de sonidos caracterizada por una relación interválica determinada entre sus elementos. Por ejemplo, la escala mayor está formada por los intervalos T-T-S-T-T-T-S, donde T es un tono y S un semitono.

escala cromática Escala que incluye todas las alturas, dado que se construye por sucesión del semitono, que es el intervalo más pequeño posible.

grado de la escala Elemento específico de una escala. Los grados de la escala se denotan con números enteros, donde $\hat{1}$ es el primer grado, $\hat{2}$ el segundo grado, y así sucesivamente.

intervalo Distancia entre dos notas.

inversión de un acorde Acorde cuya raíz no figura en el bajo, sino en una voz superior. Un acorde está en *primera inversión* si su tercera es el bajo, y en *segunda inversión* si su quinta es el bajo.

raíz de un acorde Nota de la cual se origina el acorde, por ejemplo, do en el acorde do-mi-sol.

tónica Primer grado de la escala. Ver también *acorde de tónica*.

tríada Una nota con su tercera y su quinta. Por ejemplo: do-mi-sol.

Composici3n tipogr3fica por L^AT_EX 2_ε
Generado el 11 de junio de 2014