Universidad Michoacana de San Nicolás de Hidalgo

División de Estudios de Posgrado
de la
Facultad de Ingeniería Eléctrica

CARACTERIZACIÓN E IDENTIFICACIÓN ROBUSTA DE SEÑALES DE AUDIO

TESIS DE DOCTORADO

Que para obtener el grado de
DOCTOR EN CIENCIAS EN INGENIERÍA ELÉCTRICA

Presenta:
Alain Manzo Martínez

Director de Tesis:
Doctor en Ciencias en Ingeniería Eléctrica
José Antonio Camarena Ibarrola

Morelia Michoacán, México Septiembre 2014

CARACTERIZACIÓN E IDENTIFICACIÓN ROBUSTA DE SEÑALES DE AUDIO

Los Miembros del Jurado de Examen de Grado aprueban
la **Tesis de Doctorado en Ciencias en Ingeniería Eléctrica Opción en Sistemas Computacionales** de ***Alain Manzo Martínez***

Dr. Félix Calderón Solorio
Presidente del Jurado

Dr. José Antonio Camarena Ibarrola
Director de Tesis

Dr. Juan José Flores Romero
Vocal

Dr. Mario Graff Guerrero
Vocal

Dr. Rafael González Campos
(Facultad de Ciencias Físico Matemáticas)
Revisor Externo

Dr. J. Aurelio Medina Rios
Jefe de la División de Estudios de Posgrado de la Facultad de Ingeniería Eléctrica. UMSNH
(Por reconocimiento de firmas)

UNIVERSIDAD MICHOACANA DE SAN NICOLÁS DE HIDALGO
Agosto 2014

A mis padres Reynaldo y Mary.

A mis hermanos Ivan, Ilie, Aline, Lizet y Ezequiel.

A mis sobrinos Alexander, Arlette y Camila.

AGRADECIMIENTOS

Gracias al Dr. José Antonio Camarena Ibarrola, no solo por el esfuerzo y dedicación que tuvo para dirigir y hacer posible este trabajo, sino también por sus enseñanzas y su valorable amistad ofrecida durante este tiempo.

Gracias a la Universidad Michoacana de San Nicolás de Hidalgo y en especial a la División de Estudios de Posgrado de la Facultad de Ingeniería Eléctrica por las facilidades que me otorgaron durante mis estudios.

Gracias al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo dado para realizar este trabajo.

Gracias a todos los profesores que forman parte del Departamento de Posgrado, en especial al Dr. Aurelio Medina, Dr. Claudio Fuerte, Dr. Félix Calderón, Dr. J. José Flores y Dr. Rafael Gonzáles por su apoyo y amistad durante este periodo.

Gracias a mi Familia y Amigos.

LISTA DE PUBLICACIONES

ARTICULOS DE REVISTA

1. Manzo-Martínez A. and Camarena-Ibarrola J. A. A New and Efficient Alignment Technique by Cosine Distance. *International Journal of Combinatorial Optimization Problems and Informatics*. Vol. 4, No. 1, Jan-April 2013, pp. 12-24. ISSN: 2007-1558.

CONFERENCIAS INTERNACIONALES

1. Manzo-Martínez A. and Camarena-Ibarrola J. A. A New and Efficient Alignment Technique for Evaluating Time Series Similarity. *Proceedings of the 17th International Congress on Computer Science Research*. October 2011, Morelia, México, pp. 255-266.

2. Manzo-Martínez A. and Camarena-Ibarrola J. A. A Robust Characterization of Audio Signals Using the Level of Information Content per Chroma. *Proceedings of the 11th International Symposium on Signal Processing and Information Technology*. December 2011, Bilbao, España, pp. 212-217.

3. Manzo-Martínez A. and Camarena-Ibarrola J. A. An Eigenvalues Analysis with Entropy-per-Chroma Feature. *IEEE International Autumn Meeting on Power, Electronics and Computing*. November 2013, Morelia, México, pp. 1-6.

4. Manzo-Martínez A. and Camarena-Ibarrola J. A. Use of the Entropy of a Random Process in Audio Matching Tasks. *Proceedings of the 37th International Conference on Telecommunications and Signal Processing*. July 2014, Berlin, Alemania, pp. 447-452.

5. Manzo-Martínez A. and Camarena-Ibarrola J. A. Audio-to-Audio Alignment for Performances Tracking. Sent to: *13th Mexican International Conference on Artificial Intelligence* (In peer review).

RESUMEN

En este trabajo se presentan métodos de procesamiento de señales que sirven para caracterizar e identificar audio de manera robusta. Estos métodos están direccionados para tratar dos de los problemas fundamentales de todo sistema de identificación de audio. Estos problemas son: a) el proceso de extracción de características que mejor caracteriza la señal y b) la técnica de reconocimiento que mejora la sensibilidad de la etapa de identificación de audio. Con respecto al primer problema, proponemos un proceso de extracción de características que está basado en estimar la *entropía* de los coeficientes de energía de la señal utilizando un banco de filtros adaptado a las octavas de cada una de las 12 notas musicales (*croma*). A esta nueva característica de audio se le denominó "*entropía por croma*" y sirve para resaltar el contenido armónico y melódico de una señal de audio. Esta característica tiene la peculiaridad adicional de que es robusta a ruido y a cambios dinámicos de volumen, tempo y emoción en interpretaciones musicales.

Por otra parte, el segundo problema se analiza desde la perspectiva de dos sistemas de identificación de audio diferentes. El primero es un sistema de reconocimiento de voz de palabras aisladas que nos sirve para evaluar el desempeño de nuestra *Técnica de Alineamiento por Distancia Coseno* (TADC). TADC mide la similitud (distancia) entre dos series de tiempo en base a una función similar a la que tienen las estructuras de datos de cola. Las características de TADC más importantes son: a) no requiere conocimiento previo de las series de tiempo, b) no necesita una etapa de entrenamiento y c) costo computacional lineal. En los experimentos hacemos un análisis mediante curvas ROC para evaluar la sensibilidad del sistema. Finalmente, proponemos una técnica de reconocimiento de audio que es útil para alinear dos piezas musicales en tiempo real. Esta técnica está basada en un *Proceso de Markov Parcialmente Observable* (PMPO) que permite modelar probabilísticamente características de las señales para obtener la ruta de la secuencia de estados más probable en tiempo real usando la función de estado de creencia. En los experimentos diseñamos un sistema de seguimiento de audio que utiliza una señal de referencia como partitura objetivo para equipararla con la señal de audio que se captura en tiempo real. El objetivo del sistema es contar el número de estados que son estimados correctamente mediante PMPO tomando como base la ruta óptima de la secuencia de estados generada fuera de línea con el algoritmo de Viterbi.
Palabras Clave: Alineamiento de Audio, Cromagramas, Distancia Coseno, Entropía, Markov.

ABSTRACT

In this work we discuss methods related to signal processing which are useful for characterizing and identifying audio in a robust way. These methods are addressed to discuss two of the fundamental problems in all audio identification systems. These problems are: a) the audio features extraction process which best characterizes the signal and b) the recognition technique that enhances the sensitivity of the audio identification stage. With respect to the first problem, we propose a new features extraction process which is based on estimating the *entropy* of the energy coefficients of the signal using a filters bank adapted to the octaves of each of the twelve musical notes (*chroma*). We have called this new audio feature "*entropy-per-chroma*" and it is useful to highlight the harmonic and melodic content of a signal. This feature has also the characteristic of being robust to noise and dynamic changes of volume, tempo and excitement in audio performances.

On the other hand, the second problem is analyzed from the perspective of two different audio identification systems. The first is an isolated words speech recognition system which is used to evaluate our *Alignment Technique by Cosine Distance* (ATCD). ATCD measures the similarity (distance) between two time series based on a function which is similar to that of the data structures of queue. The most important features of ATCD are: a) it does not require a-priori knowledge about the time series, b) it does not need a training stage and c) linear computational cost. In the experiments we use ROC curves to evaluate the sensitivity of the system. Finally, we propose an audio recognition technique that is useful to align two musical pieces in real time. This technique is based on a *Partially Observable Markov Process* (POMP) which allows us to model time audio features in a probabilistic way, in order to obtain the path of the most probable states sequence in real time by the belief state. In our experiments we design an audio tracking system which uses a reference signal as a target score for equating it with the incoming audio signal in real time. The system goal is to count the number of states that are correctly estimated by POMP taking as reference the optimal path of the most probable states sequence generated off-line by the Viterbi algorithm.
Keywords: Audio Alignment, Chromagrams, Cosine Distance, Entropy, Markov.

CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE SÍMBOLOS

Símbolo	Descripción
HA	Huella de Audio
TDF	Transformada Discreta de Fourier
TDC	Transformada Discreta Coseno
b	Número de bandas y número de frecuencias por octava
MFCC	Mel Frequency Cepstral Coefficients
P	Disribución de probabilidad discreta
p_k	Valores discretos de la distribución P
$H(P)$	Entropía de Shannon
v_k	Valores de amplitud de la señal de audio
I	Nivel de contenido de información de una señal
σ^2	Varianza de una función de densidad gaussiana
$p(x)$	Función de distribución de probabilidad continua
$N(0, \Sigma)$	Distribución normal con media eero
Σ	Matriz de covarianza
V_n	Volumen encerrado por $\mathbb{R}_n$
k_n	Número de muestras en $\mathbb{R}_n$
h_n	Lado de un hipercubo
$p_n(x)$	n-ésimo estimado de $p(x)$
$v(u)$	Función ventana
d	Número de valores de croma; Número de valores de entropía por croma; Dimensión de los vectores $\boldsymbol{A_0}$ y $\boldsymbol{B_0}$
TQC	Transformada Q Constante
f_0	Frecuencia inferior en Hertz
f_{max}	Frecuencia máxima en Hertz
Q	Razón constante de frecuencia a ancho de banda
$N(k)$	Longitud de ventana
f_s	Frecuencia de muestreo
$X_{QC}(k)$	Componentes de frecuencia del espectro Q constante
$x(n)$	Señal de Audio
$w_{N(k)}$	Función ventana de longitud $N(k)$
$croma(d)$	d-ésimo valor de croma
EC	Entropía por Croma
VEC	Valores de Entropía por Croma
$w(k)$	Ventana de Hann
$X(l)$	Componentes de frecuencia a partir de la TDF

Símbolo	Descripción
$f(k)$	Frecuencia de los componentes espectrales del espectro Q constante
$VEC(d)$	d-ésimo valor de entropía por croma
H_d	Entropía del d-ésimo croma
VC	Valores de Croma
HAVC	Huella de Audio de Valores de Croma
HAEC	Huella de Audio de Entropía por Croma
$h(n)$	Filtro de pre-énfasis
a	Constante del filtro de pre-énfasis
$F(d,m)$	Función de codificación para HAEC y HAVC
P_{ruido}	Potencia de la señal de ruido
$P_{señal}$	Potencia de la señal de audio
SNR	Relación señal a ruido
TPV	Tasa de predicción verdadera
TPF	Tasa de predicción falsa
$\boldsymbol{x}$ e $\boldsymbol{y}$	Series de Tiempo
$s(\boldsymbol{x},\boldsymbol{y})$	Función de similitud entre dos conjuntos o series de tiempo
$d(\boldsymbol{x},\boldsymbol{y})$	Función de distancia o distancia métrica
$\boldsymbol{x^*}$ e $\boldsymbol{y^*}$	Transformaciones de series de tiempo
μ	Media de un conjunto de datos
$\boldsymbol{W}$	Trayectoria de doblado
$DTW(\boldsymbol{x},\boldsymbol{y})$	Trayectoria óptima de doblado
$\delta(i,j)$	Trayectoria óptima
Z	Tasa de cruces por cero de una señal de audio
E	Energía de una señal de audio
sr	Señal de referencia para segmentación
k_1 y k_2	Pesos de la señal de cruces por cero y energía
$X(k)$	Transformada de Fourier de tiempo corto
$H(k,m)$	Banco de filtros de Mel
$X'(m)$	Logaritmo de la transformada de Fourier de tiempo corto
$f_c(m)$	Frecuencias centrales de los filtros de Mel
κ	Frecuencia en la escala de Mel
$c(l)$	l-ésimo MFCC
TADC	Técnica de Alineamiento por Distancia Coseno
$\boldsymbol{A}$, $\boldsymbol{B}$	Arreglos d-dimensionales
θ	Ángulo entre dos vectores, conjunto de parámetros de un modelo de mezcla de gaussianas
$\boldsymbol{A_0}$, $\boldsymbol{B_0}$	Arreglos de soporte d-dimensionales
$\boldsymbol{A_x}$, $\boldsymbol{B_x}$	Arreglos de datos alineados

Símbolo	Descripción
θ_i	Ángulo de actualización
$\theta_1, \theta_2, \phi_1, \phi_2$	Ángulos de referencia en TADC
z	Valor medio entre dos puntos
$TADC\left(\boldsymbol{x}, \boldsymbol{y}\right)$	Medida de similitud para series de tiempo de diferente longitud
w_j	Pesos de un MMG
$\phi\left(y\|\mu, \Sigma\right)$	Densidad de probabilidad gaussiana
$\varsigma_{ij}^{(m)}$	Probabilidad de que la i-ésima muestra pertenezca a
	la j-ésima componente gaussiana en la m-ésima iteración
HMM	Hidden Markov Model
S	Conjunto de estados
O	Conjunto de símbolos observables
A, a_{ij}	Matriz de transición de estados, Elementos de la matriz A
$b_i\left(o\right)$	Distribución de probabilidad de observaciones
s_i, s_j	Estado anterior, Estado actual
Π, π_i	Conjunto de probabilidades de estado inicial, probabilidad
	de que el estado inicial sea el s_i
$\lambda = (\Pi, A, B)$	Definición de un HMM
X_1^T	Secuencia de observaciones
$\alpha_i\left(t\right)$	Probabilidad de una secuencia parcial de observaciones
$\beta_i\left(t\right)$	Probabilidad de la secuencia de operaciones parcial
$Q_1^{T^*}$	Secuencia óptima de estados
$\delta_i\left(t\right)$	Probabilidad máxima de generación de una secuencia
$\varphi_i\left(t\right)$	Matriz de índices de estados más probables
q_t^*	Secuencia de estados más probables
$\gamma_i\left(t\right)$	Probabilidad de estar en el estado i en el instante t
$\xi_{ij}\left(t\right)$	Probabilidad de estar en el estado i en el instante t
	e ir al estado j en el instante $t+1$
$\pi_i^{'}$,$a_{ij}^{'}$,$b_j^{'}\left(k\right)$	Parámetros de reestimación de un HMM
$\gamma_{jk}\left(t\right)$	Probabilidad de estar en el estado j en el tiempo t con
	la k-ésima componente de la mezcla contabilizando para O_t
MDP	Markov Decision Process
POMDP	Partially Observable Markov Decision Process
Ω	Conjunto finito de observaciones
Ψ	Conjunto de probabilidades de observación
H_t	Trazo o historia de acciones y observaciones
PMPO	Proceso de Markov Parcialmente Observable

Símbolo	Descripción
$b_t(s)$	Estado de creencia
ρ	Constante de normalización
$\varphi_{ij}(a)$	Probabilidad de observar $o_j \in \Omega$ después de que la acción a es tomada
SSA	Sistema de Seguimiento de Audio
ADVP	Análisis de Descomposición de Valores Propios
λ	Valor propio
VPD	Valor Propio Dominante
$\boldsymbol{v}$	Vector propio

Capítulo 1

INTRODUCCIÓN

El procesamiento digital de señales de audio basado en contenido, es un área de investigación que juega un papel muy importante en el estudio de sistemas que tratan con voz y música. Esta área se encuentra dividida en las siguientes categorías:

- Identificación, Indexamiento y Recuperación de Audio.
- Segmentación y Discriminación de Voz o Música.
- Caracterización, Clasificación y Categorización de Audio.
- Huellas de Audio.
- Recuperación de Información de la Música.
- Minería de Datos y Semántica de la Música.

El estudio de estos tópicos ha permitido generar un amplio rango de aplicaciones tales como las desarrolladas en la industria de la música, manejo y recuperación de archivos multimedia, tratamiento de comerciales publicitarios, sistemas de seguimiento de audio, sintetizadores de voz y música, entre otras [Typke05].

Este trabajo está principalmente enfocado en los tópicos de caracterización e identificación de señales de audio. La caracterización de señales de audio es un problema general de todos los Sistemas de Identificación de Audio Basados en Contenido (SIABC). La caracterización de una señal está directamente relacionada con el proceso de extracción de características y con la forma en como ésta será representada por el SIABC [Pickens01]. Por ejemplo, a la música monofónica después de aplicarle un proceso de

extracción de características se le puede representar mediante una cadena de caracteres o símbolos, donde cada símbolo describe una nota o un par de notas consecutivas. Por otro lado, a la música polifónica se le puede representar mediante un conjunto de eventos de propiedades como inicios de nota, compases, tempo, tonos, duración, o también por un conjunto de propiedades estadísticas de la señal [Typke05].

Otros enfoques usados en SIABC son las Huellas de Audio (HA). HA usa diferentes representaciones (ej. palabras binarias, sucesiones de vectores de números reales, libro de códigos, etc.) que describen diferentes fundamentos y terminologías [Cano05a, Cano05b]. Por lo tanto, la caracterización de la señal es un tópico que debe estudiarse con cuidado ya que no solo determina la manera en cómo se representa una señal de audio, sino también la dimensionalidad a tratar, el tiempo de procesamiento del sistema y el costo en memoria.

Otro problema importante que se considera en SIABC es elegir la técnica de reconocimiento que más se adecue al tipo de caracterización y al método de búsqueda, dado que ésta se relaciona directamente con el tiempo de búsqueda del sistema. Cuando la representación del audio son sucesiones de vectores de números reales, es común usar medidas de similitud o distancias métricas [Cano05b]. Sin embargo, para el reconocimiento de conjuntos de eventos de propiedades, un modelo probabilístico es preferido. Por ejemplo, una cadena de Markov puede representar un conjunto de características melódicas como secuencias de intervalos, secuencias de tonos, secuencias de ritmo, localizaciones de compases, entre otros [Typke05].

El alineamiento de señales de audio es un problema particular presente en la etapa de reconocimiento de audio. Varias aplicaciones relacionadas con SIABC requieren soluciones eficientes a este problema. Una aplicación que se encuentra muy referenciada en la literatura es cuando una persona quiere saber el nombre de una canción o el autor de ésta y únicamente recuerda una parte de la canción o sabe como tararearla [Yu08, Kotsifakos12]. Si una persona mediante un micrófono interpreta bien el canto o el tarareo de la pieza, la aplicación le proporcionará el nombre de la canción, el artista, el álbum y los enlaces donde la puede comprar o adquirir. Este problema se le conoce como consulta por canto y/o consulta por tarareo. Otra aplicación muy común es cuando se intenta identificar una canción usando una pequeña porción del sonido grabado de la música que se reproduce en algún lugar, o cuando un compositor quiere saber si la nueva pieza musical que ha compuesto se parece en ritmo al de otras composiciones [Wang03]. Así, la búsqueda en la base de datos del sistema debe proporcionar no solo la canción a la cual pertenece el contenido de audio grabado, sino también las posibles versiones de ésta y aquellas piezas musicales que estén a una distancia relativamente

corta de este contenido.

Finalmente, se tiene el problema de alineamiento de señales de audio en tiempo real, que por lo general se encuentra en los Sistemas de Seguimiento de Audio (SSA). SSA requiere hacer el alineamiento de una pieza musical con respecto a una partitura objetivo, un archivo MIDI[1] o con respecto al audio de una interpretación diferente de la pieza musical. Para este problema se usan modelos estocásticos, programación dinámica y procesamiento de cadenas de caracteres para dar una solución parcial a este problema[Cano99, Orio03, Hanna08].

1.1. Objetivo

Investigar y estudiar tópicos matemáticos y computacionales que permitan aportar al estado del arte una nueva característica de audio que sea robusta a ruido, distorsión y variación dinámica de la señal de audio. Así mismo, proponer nuevas estrategias para alinear señales de audio que cumplan con los requerimientos necesarios para que puedan ser utilizadas en aplicaciones de tiempo real.

1.2. Justificación

En este trabajo consideramos señales de audio de voz y de interpretaciones con música monofónica y polifónica de todo tipo de género. Ésto es importante notar, ya que los trabajos que se encuentran en la literatura sobre caracterización e identificación de audio, tienen en común que direccionan la metodología de la investigación de acuerdo al tipo de señal y al proceso de extracción de características. Sin embargo, en este trabajo proponemos un proceso de extracción de características y dos técnicas de alineamiento que en conjunto permiten desarrollar diferentes trabajos relacionados con la identificación de audio tanto en línea como fuera de línea, sin importar de qué tipo de señal se trate.

1.3. Problemática

Normalmente una pieza de audio de cualquier tipo (discurso, elocución, pauta pu-

[1] *MIDI es un protocolo de comunicación serial estándar que permite a los computadores, sintetizadores, secuenciadores, controladores y otros dispositivos musicales electrónicos comunicarse y compartir información para la generación de sonidos. La información de un archivo MIDI define diversos tipos de datos como números que pueden corresponder a notas particulares, números de patches de sintetizadores o valores de controladores. Gracias a esta simplicidad, los datos pueden ser interpretados de diversas maneras y utilizados con fines diferentes a la música. El protocolo incluye especificaciones complementarias de hardware y software. Permite por ejemplo reproducir y componer música en este formato. Se caracteriza por la ligereza de los archivos, pudiendo almacenarse multitud de melodías complejas, como las de música clásica tocadas con varios instrumentos, en muy poca memoria.*

blicitaria, canción o interpretación) está expuesta a diferentes factores que corrompen los datos de la señal de audio que se encuentran almacenados en un medio digital, que se reproducen mediante un dispositivo o que son capturados por un medio electrónico en tiempo real. En los últimos años la mayoría de las investigaciones se centran en el análisis y procesamiento de los datos de las señales que son degradadas o distorsionadas. Estas investigaciones proponen soluciones eficientes para mejorar la calidad de los datos y así crear sistemas inteligentes que los interpreten para que realicen una determinada tarea o para que reproduzcan la señal con una mayor fidelidad. Sin embargo, de acuerdo al tipo de sistema o aplicación y a los factores que los corrompen, el problema de análisis y procesamiento de los datos llega a ser complejo y algunas veces intratable. Por otra parte, las señales de audio que son generadas en tiempo real como las señales de elocuciones de palabras o de las interpretaciones en vivo, agregan a estos sistemas un problema conocido como alineamiento de audio en tiempo real. Este problema se presenta porque nunca una señal de audio es igual a su réplica ya que las diferentes fuentes de variabilidad hace que el tiempo de ejecución de éstas se desplace, comprima o se expanda de manera no lineal y compleja.

En este trabajo proponemos un proceso de extracción de características de la señal de audio que es muy robusto a ruido y a la variación dinámica en tiempo y amplitud de la señal. Este proceso nos permite caracterizar los datos extraídos de la señal de forma que se puedan desarrollar sistemas de identificación de audio más eficientes. Además, para el problema de alineamiento de audio se proponen dos técnicas de alineamiento que pueden ser usadas eficientemente en sistemas de reconocimiento de voz y sistemas de seguimiento de audio.

1.4. Motivación

La motivación que tuvimos para contribuir en la problemática anterior descansó en el trabajo de Camarena-Ibarrola [Camarena09] donde introduce una huella de audio que tiene una elevada inmunidad al ruido. El proceso de extracción de la característica que utiliza esta huella se basó en extraer la *entropía espectral* por bandas de la señal. Esta característica de audio resultó ser muy robusta para diferentes clases de degradaciones de las señales de música.

Ya que nuestro objetivo no es proponer una nueva huella audio sino una característica que sea robusta a ruido y a las variaciones dinámicas de tiempo y amplitud de la señal, sólo tomamos como base el concepto de *entropía* de ese trabajo. Esto nos garantiza que nuestra característica posea elevada inmunidad al ruido, sin embargo, el reto de hacer

esta característica también robusta a las variaciones de tiempo y amplitud de la señal, fue lo que realmente nos motivó para aportar en el estado del arte.

Una motivación extra que tuvimos, fue el hecho de querer aportar nuevas estrategias para hacer alineamiento de audio en tiempo real y así desarrollar sistemas que involucren el seguimiento de audio, tales como: a) maestros virtuales de instrumentos musicales, b) sistemas de acompañamiento automático de músicos o cantantes, c) sistemas de efectos especiales en interpretaciones en vivo, entre otros. Estas estrategias se pensaron para hacer uso de la distancia Coseno y los *modelos de decisión de Markov parcialmente observables*. Estos últimos, por la capacidad que tienen de corregir el estado de un sistema ante las posibles acciones erróneas hechas en el pasado mediante las observaciones del entorno que perciben.

1.5. Estructura de la Tesis

En esta sección se presenta la estructura que tiene este trabajo. En la Figura 1.1 se muestra el diagrama a bloques general con los campos más importantes del trabajo. A continuación se describe brevemente este diagrama.

En el Capítulo 1 se da una breve introducción al tema de investigación y se proporcionan el objetivo, justificación, problemática y motivación del trabajo. Finalmente, se revisa la estructura del trabajo por capítulos y secciones.

En el Capítulo 2 se da una revisión de los fundamentos teóricos más importantes que forman parte de la investigación. El capítulo comienza con una breve introducción sobre la importancia de estos fundamentos a lo largo del trabajo. En la Sección 2.2 se exponen dos características de audio, *cromagramas* y *entropía espectral multibanda*. Ambas características son esenciales para comprender el proceso de extracción de la característica de audio presentada en el Capítulo 3. En la Sección 2.3 se exponen conceptos básicos sobre series de tiempo, funciones de distancia y se concluye con una revisión a la técnica de alineamiento temporal dinámico. Estos tópicos ayudan a comprender las bases en las que se soporta la técnica de alineamiento presentada en el Capítulo 4. Finalmente, en la Sección 2.4 se da una breve introducción a los modelos estocásticos de ventanas de Parzen, mezcla de gaussianas y modelos ocultos de Markov. Esta sección es parte fundamental del Capítulo 5 donde se expone el tema de alineamiento de señales de audio en tiempo real.

En el Capítulo 3 se presenta la característica de audio de *entropía por croma*. Se comienza con una breve introducción sobre la importancia de esta característica. En la Sección 3.2 se explica el procedimiento para obtener los *valores de entropía por croma*

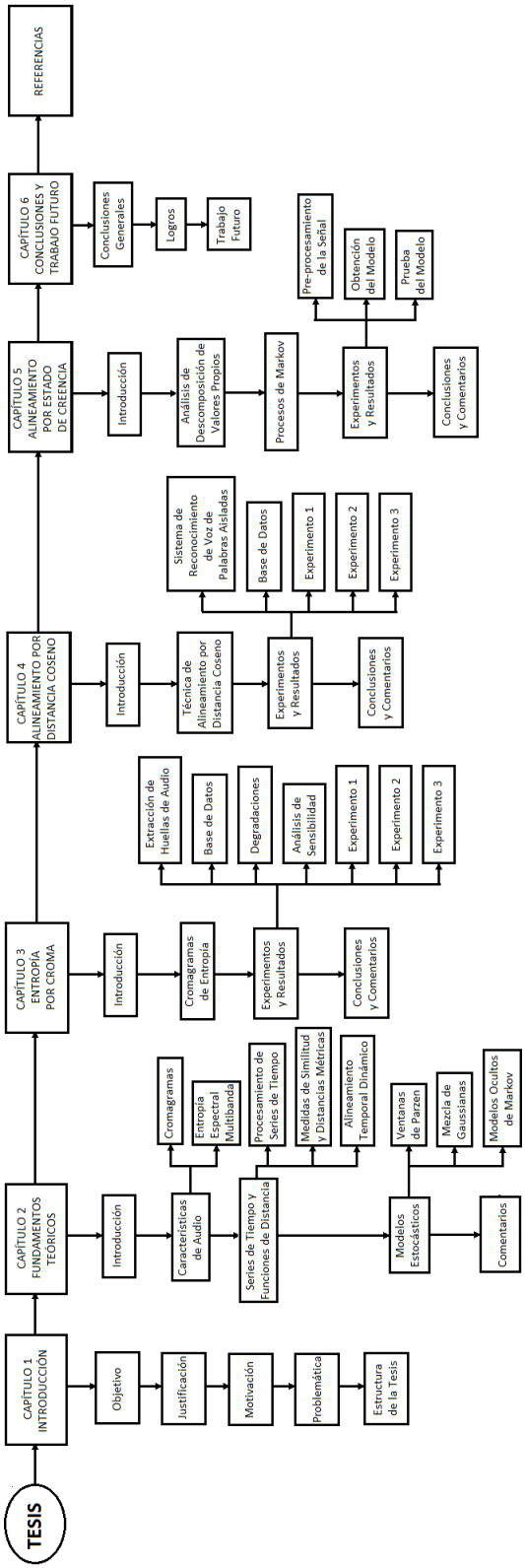

Figura 1.1: Diagrama general de la estructura de la Tesis.

que sirven para caracterizar la señal de audio mediante un *cromagrama de entropía*. Por último, se describen los dos enfoques propuestos para determinar esta característica de audio. En la Sección 3.3 se presentan los experimentos y resultados. Se inicia analizando la robustez de esta característica desde la perspectiva de huellas de audio y se presenta el procedimiento para determinar las tres huellas de audio que forman parte de los experimentos. Además, se describe la base de datos, los tipos de degradación utilizados y el análisis de sensibilidad que sirve para cuantificar el desempeño de cada una de las huellas de audio bajo prueba. Para finalizar esta sección, se presentan los resultados de los experimentos que prueban cuál de las tres huellas de audio es más robusta, tanto en música monofónica como polifónica ante presencia de ruido. Finalmente, en la Sección 3.4 se exponen las conclusiones y los comentarios del capítulo.

En el Capítulo 4 se presenta la *Técnica de Alineamiento por Distancia Coseno*. La introducción del capítulo es proporcionada en la Sección 4.1. Después, en la Sección 4.2 se da una descripción detallada de cómo se constituyen y evalúan los pasos del algoritmo de esta técnica. La sección concluye proporcionando el algoritmo completo para llevarlo fácilmente a la implementación. En la Sección 4.3 se describen los experimentos y resultados. Dentro de esta sección se explica cómo está constituido el sistema de identificación de voz de palabras aisladas que sirve para evaluar el desempeño de la técnica. Además, se describe la base de datos y los experimentos de alineamiento realizados con las señales de voz de diferentes elocuciones usando tanto alineamiento temporal dinámico como alineamiento por distancia coseno. Por último, en la Sección 4.4 se exponen las conclusiones y los comentarios de este capítulo.

En el Capítulo 5 se presenta el tema de alineamiento de audio por *estado de creencia*. En la Sección 5.1 se da una breve introducción sobre las partes de un sistema básico de seguimiento de audio. En la Sección 5.2 se describe el análisis de descomposición de valores propios que sirve para discutir la variación dinámica de tiempo y amplitud de una señal audio. En la Sección 5.3 se da revisión de los procesos de decisión de Markov ya que son fundamentales para comprender los *procesos de Markov parcialmente observables* que proponemos para alinear señales de audio. En la Sección 5.4 se presentan algunos resultados y experimentos preliminares. En esta sección se describen las características del sistema a implementar, el método de pre-procesamiento de la señal, los pasos para obtener el modelo del sistema y por último, la prueba del modelo con los algoritmos de Viterbi y *estado de creencia*. Finalmente, en la Sección 5.5 se proporcionan las conclusiones y comentarios de este capítulo.

En el Capítulo 6 se proporcionan las conclusiones generales, los logros y trabajo futuro de la investigación. Finalmente, se da el listado de las referencias del trabajo.

Capítulo 2

FUNDAMENTOS TEÓRICOS

En este capítulo se da una revisión a los fundamentos teóricos en los cuales se soporta este trabajo de investigación. Este capítulo se encuentra dividido en cuatro secciones principales. La Sección 2.1 ayuda a que el lector se familiarice con la temática y problemática de la investigación. La Sección 2.2 introduce al lector en los conceptos de *cromagrama* y *entropía espectral multibanda* que son necesarios para comprender el proceso de extracción de la *entropía por croma* que se presenta en el Capítulo 3. La Sección 2.3 está enfocada a guiar al lector en el entendimiento de los tópicos sobre procesamiento y funciones de distancia de series de tiempo que sirven de soporte para la *técnica de alineamiento por distancia coseno* que se introduce en el Capítulo 4. Por último, la Sección 2.4 introduce al lector en modelos estocásticos. Se da un panorama general sobre ventanas de Parzen, modelo de mezcla de gaussianas y modelos ocultos de Markov, todos ellos relacionados con el tema de alineamiento de audio que se expone en el Capítulo 5.

2.1. Introducción

La caracterización de señales de audio se relaciona con el proceso de extracción de las características que describen de manera abstracta una señal y reflejan sus aspectos de percepción más relevantes. Para extraer las características de una grabación de voz o música es común segmentar la señal de ésta en tramas cortas, posiblemente traslapadas lo suficientemente cerca entre sí, de tal manera que no se cubran múltiples eventos distinguibles o perceptuales en una sola trama [Aucouturier07]. Wold et al. [Wold96] listan algunas características que son comúnmente extraídas a partir de tramas de audio con diferentes duraciones.

La mayoría de los sistemas de identificación de audio utilizan el proceso de extracción

de características acompañado de técnicas eficientes de reconocimiento para identificar las señales de las grabaciones de audio. Estas técnicas encuentran las grabaciones que suenan similar al audio que se consulta en estos sistemas. Si las señales de audio de estas consultas son segmentos de corta duración, por lo general, una función de distancia o un enfoque de huellas de audio es usado para identificarlas. Las huellas de audio utilizan dos procesos fundamentales para ser determinadas, un proceso de extracción de características y un proceso de modelado; este último se refiere al tipo de representación que describe en forma compacta una señal, de manera que sea lo más robusta posible contra degradaciones típicas del audio.

Generalmente las huellas de audio trabajan muy bien en estos sistemas, pero el problema se complica cuando se quiere identificar diferentes versiones de la grabación de audio (ej. una pieza musical ejecutada por diferentes músicos e instrumentación). Este problema por lo regular lleva a una representación de la señal en series de tiempo, cadenas de caracteres o conjuntos de eventos de propiedades. De esta manera, se pueden aplicar técnicas de alineamiento o modelos estocásticos para tratar las variaciones dinámicas de tiempo y amplitud de las señales de las diferentes versiones de la grabación.

2.2. Características de Audio

En esta sección se presentan dos características de audio que son necesarias en este trabajo. En principio se discute el proceso para extraer de una señal la característica de audio basada en *cromas*. Posteriormente, se discuten algunos conceptos básicos sobre *entropía* y se finaliza con el proceso de extracción de la característica de audio basada en la *entropía espectral multibanda*.

2.2.1. Cromagramas

En el estudio de la música es común usar varias representaciones para entender su comportamiento, tanto temporal como en frecuencia. Una de estas representaciones son los *cromagramas*. Un *cromagrama* es una sucesión de vectores que describen la distribución de la energía relativa sobre clases tradicionales de tonos y semitonos codificados por los atributos C, C#, D, ... , B (es decir, las notas musicales de Do, Re, Mi, ... , Si). De acuerdo a Shepard [Shepard64] la sensación de un tono musical puede ser caracterizada mediante dos dimensiones, la altura de tono y el *croma*. La altura de tono describe el incremento general en el tono de un sonido conforme su frecuencia incrementa. La dimensión de la altura de tono es particionada en octavas musicales. El *croma*, por otro

lado, es de naturaleza cíclica con periodicidad de octava. Bajo esta formulación, dos tonos separados por un número entero de octavas comparten el mismo *valor de croma*. El rango del *croma* está generalmente dividido en 12 clases de tonos, donde cada clase de tono corresponde a una de las doce notas musicales. Por ejemplo, la clase C contiene todos los atributos C de todas las posibles octavas (C0, C1, C2, ...). Los tonos (notas musicales) de la misma clase comparten el mismo *croma* y producen una sensación auditiva similar.

En resumen, un *cromagrama* captura la información de los tonos para representar el contenido armónico y melódico de una señal de audio. Cada vector del *cromagrama* tiene d componentes, donde cada componente representa un *valor de croma* (VC). Los VC están relacionados con la circularidad percibida de los tonos de un sonido a partir de una octava a la otra. En conclusión, los VC al parecer son ideales para aplicaciones con música monofónica, ya que intentan representar las notas musicales y los cambios de dinámica de la música. Hay diferentes alternativas que se pueden encontrar en la literatura sobre como determinar los VC; como referencia se tiene el trabajo de Stein et al. [Stein09], donde hacen una evaluación y comparación detallada de diferentes métodos para determinarlos.

Un método común para extraer los VC de una señal de audio, es a partir de la Transformada Q Constante (TQC) [Bello05, Fitzgerald06, Zhu06, Graziosi04, Brown92]. Para extraer esta característica de audio se deben determinar las frecuencias de los componentes espectrales sobre el espectro Q constante a partir de la ecuación (2.1) dada una frecuencia inferior f_0,

$$f(k) = f_0 2^{k/b} \quad \forall k = 0, 1, ..., K-1 \tag{2.1}$$

donde b denota el número de componentes espectrales en una octava. Dado el valor de b, el número de componentes espectrales K sobre el espectro Q constante está determinado por la ecuación (2.2),

$$K = b \times log_2 \left(\frac{f_{max}}{f_0} \right) \tag{2.2}$$

donde f_{max} es la frecuencia máxima dada. La razón constante de frecuencia a ancho de banda (resolución en frecuencia) conocida como Q está dada por la ecuación (2.3).

$$Q = \left(2^{1/b} - 1\right)^{-1} \tag{2.3}$$

El ancho de banda de cada banda de frecuencia es obtenido eligiendo una ventana

de análisis $\omega(k)$, donde su longitud $N\,(k)$, está en función del índice k. De esta manera, $N(k)$ se calcula a través de la ecuación (2.4),

$$N\,(k) = Q\frac{f_s}{f\,(k)} \tag{2.4}$$

donde f_s es la frecuencia de muestreo. La TQC se determina mediante la ecuación (2.5),

$$X_{QC}\,(k) = \frac{1}{N\,(k)} \sum_{n=0}^{N(k)-1} \omega\,(n,k)\,x\,(n)\,e^{\frac{-j2\pi nQ}{N\,(k)}} \tag{2.5}$$

donde $x\,(n)$ es una trama de la señal de audio y $\omega(n,k)$ es una función ventana de longitud $N\,(k)$. A partir de $X_{QC}\,(k)$, los VC se pueden obtener por medio de la ecuación (2.6),

$$croma\,(d) = \sum_{m=0}^{M} |X_{QC}\,(d+mb)| \tag{2.6}$$

donde d denota el número de *croma* y M, el número total de octavas en el espectro Q constante.

Con el método anterior se puede obtener un vector del *cromagrama* por trama de la señal. Una sucesión de vectores con VC extraída de un segmento de audio se le conoce como *cromagrama*. Un *cromagrama* se puede representar mediante una imagen de d renglones y un número de columnas que depende de la duración del segmento. El eje horizontal de la imagen representa el tiempo en relación al número de tramas, el eje vertical representa el *croma* y los niveles de gris representan la energía de cada *croma* y trama de la señal. En la Figura 2.1 se observa el *cromagrama* de una señal de audio con 10 segundos de duración. En el capítulo siguiente se retoma este tema ya que forma parte del desarrollo y los resultados de ese capítulo.

2.2.2. Entropía espectral multibanda

En el ámbito de la teoría de la información, la *entropía* de Shannon se relaciona con la incertidumbre de una fuente de información [Shannon48]. Por ejemplo, la *entropía* se usa para medir la predictibilidad de una señal aleatoria, la cantidad de "ruido" o "desorden" que contiene o lidera un sistema y la "picudez" de una función de distribución de probabilidad [Misra04]. En general, se puede decir que la *entropía* es exitosamente aplicada a modelos de la información, pero suena inusual que ésta se utilice como una característica de una señal para la caracterización de audio. Sin embargo, Hemant et al.

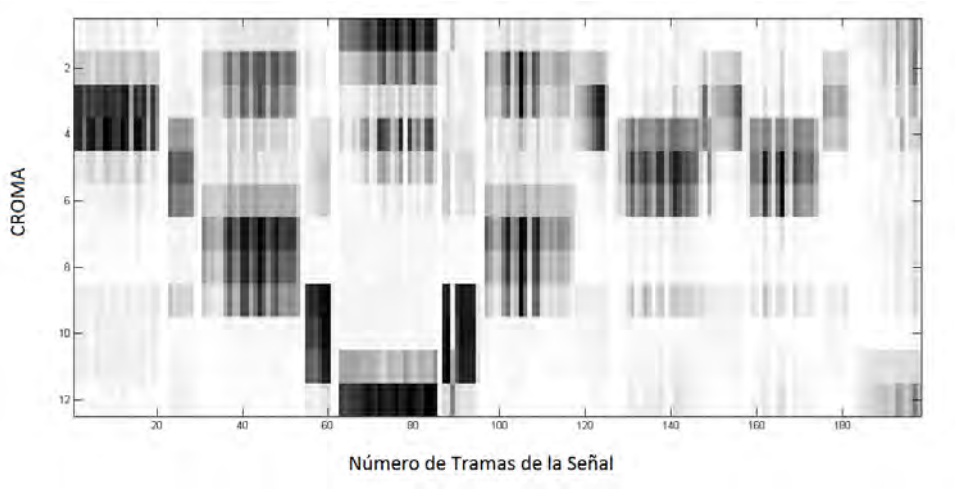

Figura 2.1: *Cromagrama* de una señal de audio de música monofónica.

[Misra04, Misra05] fueron los primeros en introducir el concepto de *entropía espectral* como una característica de audio para el reconocimiento de voz automático.

Basándose en la idea de Hemant, Camarena-Ibarrola [Camarena06, Camarena09, Camarena10] introduce la característica de audio de *entropía espectral multibanda* que consiste en agrupar coeficientes de energía por bandas para calcular la *entropía* en cada banda. Antes de exponer esta característica formalmente se revisará la definición de *entropía* para una variable aleatoria discreta y continua.

2.2.2.1. *Entropía de una variable aleatoria*

Sea $P = (p_1, p_2, ..., p_n)$ una distribución de probabilidad discreta finita, por lo tanto, $p_k \geq 0$ para $k = 1, 2, ..., n$, y $\sum_{k=1}^{n} p_k = 1$. Considere un experimento con n posibles resultados, cada uno con probabilidad $p_1, p_2, ..., p_n$. La cantidad de incertidumbre del resultado del experimento es usualmente medida por la *entropía de Shannon*, $H(P)$, que se determina a partir de la ecuación (2.7) [Shannon48].

$$H(P) = -\sum_{k=1}^{n} p_k ln\,(p_k) \tag{2.7}$$

La *entropía* de P puede ser interpretada no sólo como una medida de incertidumbre sino también como el nivel de contenido de información de una señal. Sea $v_1, v_2, ..., v_n$ los valores de amplitud posibles de las muestras de una señal de audio. Si por ejemplo, cada muestra tiene un tamaño de palabra de 8 bits, cada muestra podría ser un entero en el rango de $[-128, 127]$. Cada v_k tiene probabilidad p_k de ocurrir, por lo tanto, $p_1, p_2, ..., p_n$

es una función de densidad de probabilidad discreta.

El nivel de contenido de información I en un valor v_k, depende únicamente de su probabilidad p_k, para $k = 1, ..., n$, tal como se expresa en la ecuación (2.8).

$$I(p_k) = -ln(p_k) \tag{2.8}$$

La *entropía* de una señal es el nivel de contenido de información esperado en la señal, por lo tanto, la *entropía* es el promedio de todos los niveles de contenidos de información ponderados por sus probabilidades de ocurrir [Rényi61]. De esta manera, la *entropía* de una señal se determina por la ecuación (2.7).

La *entropía* de una señal es una medida de que tan predecible es ésta. Si la señal es constante en un valor fijo i, entonces su distribución de probabilidad es una delta de Kronecker localizada en i, esto es,

$$p_i = \delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

y su *entropía* es igual a cero, tal como se muestra la ecuación (2.9) (note que cuando $p \to 0$, el límite de $H(p) = pln(p)$ es igual a cero).

$$H(p_i) = -\sum_{j=1}^{n} \delta_{ij} ln(\delta_{ij}) = -ln(1) = 0 \tag{2.9}$$

En el caso opuesto, si la señal tiene una distribución uniforme entonces la *entropía* será máxima, esto es, si $p_k = 1/n$ para n posibles valores, entonces la *entropía* es igual a $ln(n)$, tal como se observa en la ecuación (2.10).

$$H(p_k) = -\sum_{k=1}^{n} \frac{1}{n} ln\left(\frac{1}{n}\right) = -ln\left(\frac{1}{n}\right) = ln(n) \tag{2.10}$$

Ahora considere el caso de la *entropía* de una señal o variable aleatoria continua x asociada con una función de densidad gaussiana con media de cero y varianza σ^2 que se puede expresar mediante (2.11).

$$p(x) = \frac{e^{-\frac{x^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} \tag{2.11}$$

Tomando ciertas precauciones, la definición de la ecuación (2.7) se puede cambiar para calcular la *entropía* de la variable aleatoria continua por medio de la ecuación (2.12).

$$H\left(p\left(x\right)\right) = -\int_{-\infty}^{\infty} p\left(x\right) ln\left[p\left(x\right)\right] \tag{2.12}$$

Sustituyendo la ecuación (2.11) en (2.12), la *entropía* de una variable aleatoria con distribución gaussiana es determinada mediante la ecuación (2.13).

$$H\left(p\left(x\right)\right) = -\int_{-\infty}^{\infty} \frac{e^{-\frac{x^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} ln\left(\frac{e^{-\frac{x^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma}\right) dx = \frac{1}{2} ln\left(2\pi e\right) + \frac{1}{2} ln\left(\sigma^2\right) \tag{2.13}$$

Para el caso d-dimensional, la *entropía* H de un vector sobre un proceso aleatorio usando distribución $N\left(0, \Sigma\right)$ se determina con la ecuación (2.14), donde Σ es la matriz de covarianzas de tamaño $d \times d$ [Mohammad94].

$$H = \frac{d}{2} ln\left(2\pi e\right) + \frac{1}{2} ln\left(\left|\Sigma\right|\right) \tag{2.14}$$

En la siguiente sección se revisa el uso de la ecuación (2.14) en el proceso de extracción de la *entropía espectral multibanda*.

2.2.2.2. *Proceso de extracción de la entropía espectral multibanda*

Los pasos para determinar la *entropía espectral multibanda* de una señal de audio se presentan a continuación:

- La señal de audio se convierte en una señal monoaural promediando ambos canales (sólo si son señales estéreo).
- La señal se divide en tramas de duración de 370 ms cada una o en su defecto cada trama contiene 16,317 datos de la señal si la frecuencia de muestreo es de 44100 muestras por segundo. El traslape entre tramas es del 50 %.
- Aplicar una ventada de Hann a los datos de cada trama.
- Calcular la Transformada Discreta de Fourier de cada trama.
- Agrupar los coeficientes de energía en bandas usando las primeras 24 bandas criticas de Bark.
- Por último, calcular la *entropía* mediante la ecuación 2.14 en cada una de las bandas.

Por cada trama de la señal de audio se obtiene un vector con 24 valores de *entropía*. Una sucesión de estos vectores hace una matriz de 24 renglones con un número de columnas que depende de la duración de la señal. Tal matriz puede ser mostrada como una imagen, donde el eje horizontal representa el tiempo en relación al número de tramas, el eje vertical representa la frecuencia y los niveles de gris representan el nivel de *entropía* para cada banda y trama. A esta imagen se le conoce como *entropigrama* [Camarena09]. En la Figura 2.2 se observa un *entropigrama* de una señal de audio de duración de 5 segundos.

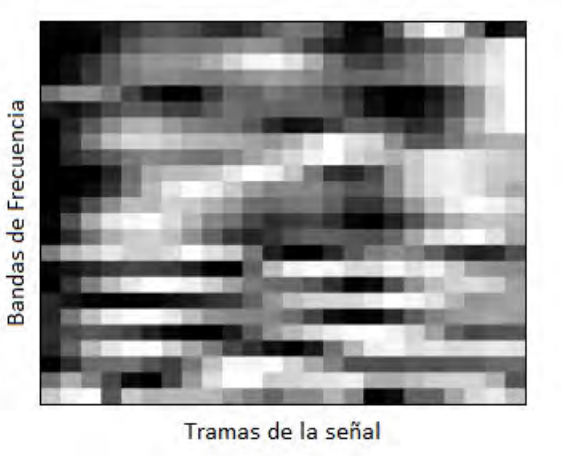

Figura 2.2: *Entropigrama* de un segmento de música polifónica de 5 segundos de audio [Camarena09].

2.3. Series de Tiempo y Funciones de Distancia

En esta sección se describen los conceptos sobre series de tiempo y funciones de distancia. En principio se introducen conceptos básicos y transformaciones de series tiempo. Posteriormente, se da una revisión a las medidas de similitud y distancias métricas que son necesarias para el desarrollo de este trabajo. Finalmente, se da un repaso a la técnica de alineamiento temporal dinámico, que será fundamental para validar algunos resultados más adelante.

2.3.1. Procesamiento de series de tiempo

Una serie de tiempo es una secuencia de números reales que representan las mediciones de una variable real en intervalos de tiempo iguales [Gunopulos00]. El análisis estadístico de series de tiempo incluye reconocimiento de patrones (ej. análisis de la

tendencia, análisis de estacionalidad, modelos de autocorrelación y autoregresivos) y previsión o pronóstico. A partir de la perspectiva de base de datos, los problemas importantes relacionados a series de tiempo son:

- ***Problema de similitud***. Usando los datos de las series de tiempo, el problema consiste en determinar si diferentes series de tiempo tienen conducta similar. Precisamente, dadas dos series de tiempo $\boldsymbol{x} = x_1, x_2, ..., x_m$ e $\boldsymbol{y} = y_1, y_2, ..., y_m$, ¿cómo se puede definir y determinar la distancia o la similitud entre ellas?
- ***Problema de Indexamiento***. Este problema consiste en encontrar el elemento más parecido a una serie que se consulta en una gran base de datos. Una solución obvia es recuperar y examinar cada secuencia en la base de datos. Sin embargo, este método no escala a grandes bases de datos. Este problema involucra el indexamiento de series de tiempo.
- ***Problema de similitud de una subsecuencia***. Este problema puede ser descrito como sigue: dada una plantilla o una consulta Q, una base de datos de referencia y una medida de distancia, encontrar la localidad que mejor iguala a Q. Por ejemplo, encontrar los otros días cuando el capital tuvo movimientos similares al de hoy.
- ***Problema de agrupamiento***. Agrupar series de tiempo es encontrar agrupamientos naturales de series de tiempo en una base de datos bajo alguna medida de similitud. Debido a la estructura única de las series de tiempo, la mayoría de los algoritmos de agrupamiento no trabajan bien para datos de series de tiempo.
- ***Problema de descubrimiento de regla***. El problema de descubrimiento de regla, es el problema de encontrar reglas relacionando patrones en una serie de tiempo a otros patrones en la misma serie de tiempo, o patrones en una serie de tiempo a patrones en otra serie de tiempo. Por ejemplo, encontrar reglas tal como "si el capital X está creciendo e Y decae, entonces Z crecerá al día siguiente".

Sea $\boldsymbol{x}$ e $\boldsymbol{y}$ dos series de tiempo, y $d(\boldsymbol{x}, \boldsymbol{y})$ la distancia entre $\boldsymbol{x}$ e $\boldsymbol{y}$. La medida de distancia para series de tiempo tiene que seguir las siguientes propiedades:

- $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$ (simetría).

- $d(\boldsymbol{x}, \boldsymbol{x}) = 0$ (constancia o autosimilitud).
- $d(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ y $d(\boldsymbol{x}, \boldsymbol{y}) = 0$ si y sólo si $\boldsymbol{x} = \boldsymbol{y}$ (positividad).
- $d(\boldsymbol{x}, \boldsymbol{y}) \leq d(\boldsymbol{x}, \boldsymbol{z}) + d(\boldsymbol{z}, \boldsymbol{y})$(desigualdad del triángulo).

Para la mayoría de las aplicaciones, la distorsión de los datos de una serie de tiempo debe ser removida. Existen varias transformaciones para remover distorsiones en datos de series de tiempo [Guojun07]. A continuación, $\boldsymbol{x} = x_1, x_2, ..., x_d$ e $\boldsymbol{y} = y_1, y_2, ..., y_d$, denotarán dos series de tiempo puras y $\boldsymbol{x}^*$ e $\boldsymbol{y}^*$ las series de tiempo resultantes de una transformación.

La transformación de desplazamiento remueve la línea media de los datos de una serie de tiempo. Precisamente, la serie de tiempo transformada $\boldsymbol{x}^* = x_1^*, x_2^*, ..., x_d^*$ está dada por la ecuación (2.15)

$$x_j^* = x_j - \mu_x, \quad j = 1, 2, ..., d$$

$$\boldsymbol{x}^* = \boldsymbol{x} - \mu_x \tag{2.15}$$

donde $\mu_x = \frac{1}{d}\sum_{j=1}^{d} x_j$.

La transformación de desplazamiento puede remover la línea media, pero ésta no puede remover la amplitud de las series de tiempo. Para remover la amplitud de las series de tiempo, se puede usar la transformación de amplitud que está dada por la ecuación (2.16)

$$x_j^* = \frac{x_j - \mu_x}{\sigma_x}, \quad j = 1, 2, ..., d$$

$$\boldsymbol{x}^* = \frac{\boldsymbol{x} - \mu_x}{\sigma_x} \tag{2.16}$$

donde $\sigma_x = \left(\frac{1}{d}\sum_{j=1}^{d} (x_j - \mu_x)^2\right)^{1/2}$.

2.3.2. Medidas de similitud y distancias métricas

Una medida o coeficiente de similitud indica la fuerza de la relación entre dos puntos de datos [Everrit93]. Entre más grande sea la semejanza entre estos dos, mayor será la

medida de similitud. Sea $\boldsymbol{x} = (x_1, x_2, ..., x_d)$ e $\boldsymbol{y} = (y_1, y_2, ..., y_d)$ dos puntos de datos d-dimensionales. La medida de similitud entre $\boldsymbol{x}$ e $\boldsymbol{y}$ será alguna función de sus valores de sus atributos, es decir,

$$s(\boldsymbol{x}, \boldsymbol{y}) = s(x_1, x_2, ..., x_d, y_1, y_2, ..., y_d)$$

La similitud es usualmente simétrica, es decir, $s(\boldsymbol{x}, \boldsymbol{y}) = s(\boldsymbol{y}, \boldsymbol{x})$. Por otra parte, una métrica es una función de distancia f definida como un conjunto E que satisface las siguientes cuatro propiedades [Anderberg73, Zhang03]:

- No negatividad: $f(\boldsymbol{x}, \boldsymbol{y}) \geq 0$;
- Reflexividad: $f(\boldsymbol{x}, \boldsymbol{y}) = 0 \Longleftrightarrow \boldsymbol{x} = \boldsymbol{y}$;
- Simetría: $f(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{y}, \boldsymbol{x})$;
- Desigualdad del Triángulo: $f(\boldsymbol{x}, \boldsymbol{y}) \leq f(\boldsymbol{x}, \boldsymbol{z}) + f(\boldsymbol{y}, \boldsymbol{z})$,

donde $\boldsymbol{x}$, $\boldsymbol{y}$, y $\boldsymbol{z}$ son puntos de datos arbitrarios. Por otro lado, una función de similitud se refiere a una función $s(\boldsymbol{x}, \boldsymbol{y})$ medida en cualquiera de los dos puntos de datos en un conjunto que satisface las siguientes propiedades:

- $0 \leq s(\boldsymbol{x}, \boldsymbol{y}) \leq 1$,
- $s(\boldsymbol{x}, \boldsymbol{x}) = 1$,
- $s(\boldsymbol{x}, \boldsymbol{y}) = s(\boldsymbol{y}, \boldsymbol{x})$,

donde $\boldsymbol{x}$ e $\boldsymbol{y}$, son dos puntos de datos arbitrarios en el conjunto. A continuación, se presentan las medidas de similitud y distancias métricas que se utilizan a lo largo de este trabajo.

La distancia Euclidiana es probablemente la distancia más comúnmente usada para datos numéricos. Para dos puntos de datos $\boldsymbol{x}$ e $\boldsymbol{y}$ en un espacio d-dimensional, la distancia Euclidiana entre ellos está definida por la ecuación (2.17)

$$d(\boldsymbol{x}, \boldsymbol{y}) = \left[\sum_{j=1}^{d} (x_j - y_j)^2\right]^{\frac{1}{2}} = \left[(\boldsymbol{x} - \boldsymbol{y})^T (\boldsymbol{x} - \boldsymbol{y})\right]^{\frac{1}{2}} \tag{2.17}$$

donde x_j e y_j son los valores del j-ésimo atributo de $\boldsymbol{x}$ e $\boldsymbol{y}$, respectivamente.

La distancia Euclidiana y Manhattan son casos particulares de la distancia Minkowsky definida por la ecuación (2.18)

$$d(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{j=1}^{d} |x_j - y_j|^r \right)^{\frac{1}{r}}, \quad r \geq 1 \tag{2.18}$$

donde r es el orden de la distancia. Note que si $r = 1$ y $r = 2$, se obtiene la distancia Manhattan y Euclidiana, respectivamente.

La idea de similitud es más consistente si se considera la definición de la distancia Hamming. Sea $\boldsymbol{x}$ e $\boldsymbol{y}$ dos secuencias binarias de misma longitud, K, la distancia Hamming entre ellas es el número de símbolos o bits en que difieren, siendo $W(\cdot)$ el peso de Hamming o el número de símbolos diferentes. La distancia Hamming está dada mediante la ecuación (2.19)

$$s(\boldsymbol{x}, \boldsymbol{y}) = W(\boldsymbol{x} \oplus \boldsymbol{y}) \tag{2.19}$$

donde $\oplus$ es la operación suma módulo 2 entre las secuencias.

La medida de similitud Coseno o distancia Coseno [Salton83] es adoptada para medir la similitud entre una transacción de datos. Sean $\boldsymbol{x}$ e $\boldsymbol{y}$ dos puntos de datos representados por un vector d-dimensional, la distancia Coseno entre $\boldsymbol{x}$ e $\boldsymbol{y}$ está dada por la ecuación (2.20)

$$s(\boldsymbol{x}, \boldsymbol{y}) = \frac{\langle \boldsymbol{x} | \boldsymbol{y} \rangle}{\|\boldsymbol{x}\|_2 \|\boldsymbol{y}\|_2} \tag{2.20}$$

donde $\langle \boldsymbol{x} | \boldsymbol{y} \rangle$ denota el producto punto, es decir,

$$\langle \boldsymbol{x} | \boldsymbol{y} \rangle = \sum_{j=1}^{d} x_j y_j$$

y $\|\cdot\|_2$ denota un vector norma, esto es,

$$\|x\|_2 = \sqrt{\sum_{j=1}^{d} x_j^2}$$

2.3.3. Alineamiento temporal dinámico

Múltiples series de tiempo pueden ser generadas a partir de la observación de sistemas biológicos, procesos estocásticos, patrones de conducta, entre otros. Independientemente

del medio por el cual se hayan generado las series de tiempo, en todas ellas por lo general, están presentes diferentes fuentes de variabilidad. Estas fuentes repercuten directamente sobre el eje del tiempo, desplazándolo, comprimiéndolo y expandiéndolo en una manera compleja y no lineal. Estas características dan origen a series de tiempo con secuencias de diferentes longitudes. La diferencia de longitudes no permite usar directamente las medidas de similitud o distancias métricas antes mencionadas. Para este problema se utilizan técnicas de alineamiento para medir la similitud entre series de tiempo.

La técnica de Alineamiento Temporal Dinámico (o DTW por sus siglas en inglés de Dynamic Time Warping) es una técnica que encuentra un alineamiento óptimo entre dos secuencias (dependientes del tiempo) bajo ciertas restricciones. Inicialmente, las secuencias están deformadas en una forma no lineal para equipararse una con la otra, esto puede observarse en la Figura 2.3.

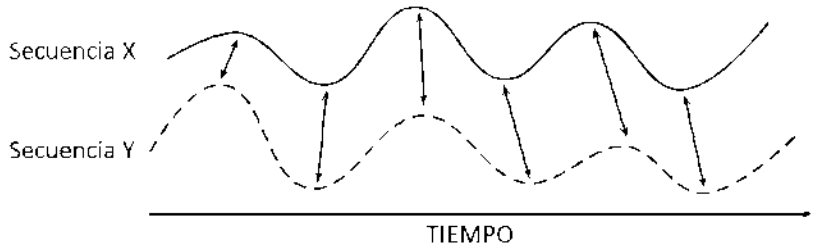

Figura 2.3: Alineamiento de dos secuencias dependientes del tiempo. Los puntos alineados son indicados por flechas [Muller07].

El objetivo de DTW es comparar dos secuencias. Estas secuencias pueden ser señales discretas, series de tiempo, o más generalmente, secuencias de características muestreadas en puntos equidistantes en tiempo. Matemáticamente, suponga que se tienen dos series de tiempo, $\boldsymbol{x} = x_1, x_2, ..., x_r$ e $\boldsymbol{y} = y_1, y_2, ..., y_s$, donde r y s no son necesariamente iguales. Para comparar las dos series de tiempo se necesita una medida de costo local, algunas veces referida como medida de distancia local. Típicamente, la distancia entre dos puntos $d(x_i, y_i)$ es pequeña si x_i e y_i son similares el uno al otro, de lo contrario $d(x_i, y_i)$ es grande. Evaluando la medida de costo local para cada par de los elementos de las series de tiempo $\boldsymbol{x}$ e $\boldsymbol{y}$, se obtiene una matriz de costo $M \in \mathbb{R}^{r \times s}$ definida mediante $M(r, s) := m(x_r, y_s)$. La Figura 2.4 muestra un ejemplo de una matriz de costo representada mediante una imagen en escala de grises. Si la distancia entre dos puntos x_i e y_j de la serie de tiempo es pequeña, el nivel de gris será cercano al color negro, mientras que para una distancia grande el nivel de gris se correrá al color blanco.

Intuitivamente, el alineamiento óptimo entre las dos series de tiempo se lleva a cabo a lo largo de un "valle" de bajo costo dentro de la matriz de costo, esto queda ilustrado

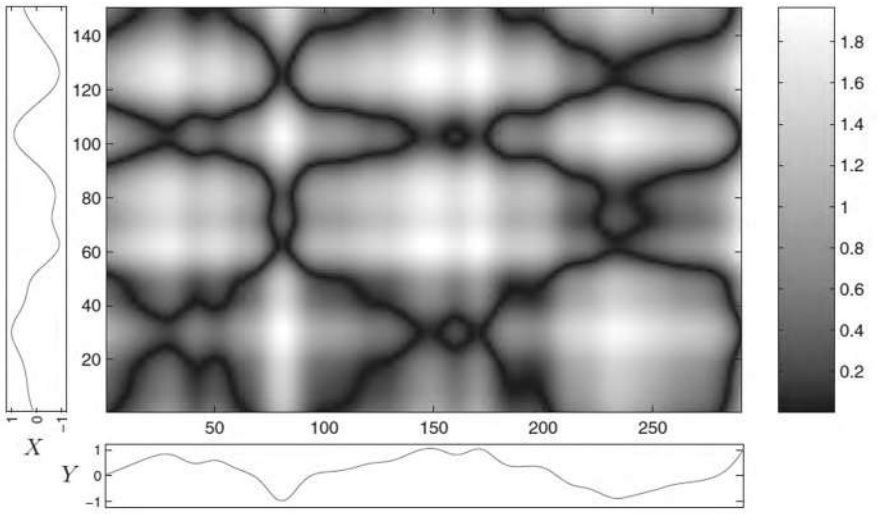

Figura 2.4: Matriz de costo de dos series de tiempo $\boldsymbol{x}$ e $\boldsymbol{y}$ usando distancia Manhattan como medida de costo local [Muller07].

en la Figura 2.5.

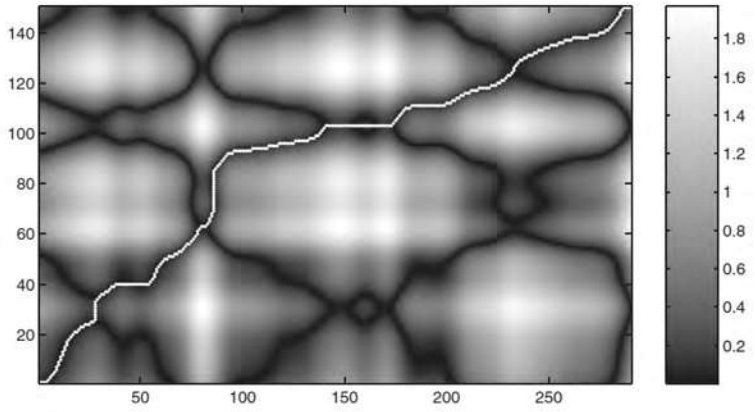

Figura 2.5: Trayectoria de alineamiento o de doblado óptimo en la matriz de costo [Muller07].

|Para encontrar el alineamiento óptimo sea M la matriz de costo de tamaño $r \times s$ con el i,j-ésimo elemento conteniendo la distancia $d(x_i, y_i)$ entre dos puntos x_i e y_i. Cada elemento (i,j) en M corresponde al alineamiento entre los puntos x_i e y_i. Entonces cada alineamiento posible entre $\boldsymbol{x}$ e $\boldsymbol{y}$ puede ser representado mediante una trayectoria de doblado (o trayectoria deformada) en la matriz M, donde una trayectoria de doblado es un conjunto de elementos contiguos de la matriz (ver Figura 2.5).

Sea $W = w_1, w_2, ..., w_K$ una trayectoria de doblado, donde el k-ésimo elemento $w_k =$

(i_k, j_k). Entonces, el $max\{r, s\} \leq K < r + s - 1$. Las trayectorias de doblado tienen algunas restricciones:

- ***Monotonicidad***. Dado $w_k = (i, j)$, entonces $w_{k-1} = \left(i^{'}, j^{'}\right)$, donde $i^{'} \leq i$ y $j^{'} \leq j$. Esto asegura que la trayectoria de doblado W no vaya hacia atrás.
- ***Continuidad***. Dado $w_k = (i, j)$, entonces $w_{k-1} = \left(i^{'}, j^{'}\right)$, donde $i \leq i^{'} + 1$ y $j \leq j^{'} + 1$. Esto restringe la trayectoria de doblado a celdas adyacentes y asegura que no se puedan omitir elementos en una secuencia.
- ***Condiciones de Frontera***. El primero y el último elemento en W son fijos, es decir, $w_1 = (1, 1)$ y $w_K = (r, s)$. Si una ventana de doblado es especificada para trayectorias de doblado, entonces solo los elementos de la matriz (i, j) con $|i - j| \leq \beta$ son considerados, donde β es el tamaño de la ventana de doblado.

Hay muchas trayectorias de doblado que satisfacen las condiciones anteriores. Una trayectoria óptima de doblado, es aquella que minimiza el costo de la trayectoria, el cual está definido por la ecuación (2.21)

$$DTW(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sqrt{\sum_{l=1}^{K} w_l}}{K} = \frac{\sqrt{\sum_{l=1}^{K} d\left(x_{i_l}, y_{i_l}\right)}}{K} \tag{2.21}$$

donde $(i_l, j_l) = w_l$ para $l = 1, 2, ..., K$. Para determinar la trayectoria óptima de doblado, se pueden generar todas las trayectoria de doblado posibles entre las series de tiempo $\boldsymbol{x}$ e $\boldsymbol{y}$. Tal procedimiento, sin embargo, podría conducir a una complejidad computacional $O(rs)$ que es exponencial en las longitudes r y s [Muller07].

La trayectoria óptima puede ser encontrada eficientemente por programación dinámica. Sea $\delta(i, j)$ la distancia de doblado de tiempo dinámico entre las subsecuencias $x_1, x_2, ..., x_i$ e $y_1, y_2, ..., y_j$. Entonces la trayectoria óptima puede ser encontrada usando programación dinámica evaluando la recurrencia dada por la ecuación (2.22).

$$\delta(i, j) = d(x_i, y_i) + min\{\delta(i - 1, j), \delta(i - 1, j - 1), \delta(i, j - 1)\} \tag{2.22}$$

La ecuación anterior acumula medidas de distancia local en cada recursión, es decir, la distancia de doblado $\delta(i, j)$ suma la distancia del elemento (i, j) con la mínima distancia acumulada hasta ese momento en las celdas adyacentes que están hacia abajo,

a la izquierda y a la derecha del elemento (i, j), esto se hace comenzando por el elemento (r, s) de la matriz de costo.

Para mejorar la sensibilidad de la técnica, se proponen funciones de restricciones globales. Tales restricciones no sólo incrementan la velocidad computacional de DTW sino también evita alineamientos patológicos mediante el control global de la ruta de una trayectoria de doblado. Más precisamente, sea $R \subseteq [1:r] \times [1:s]$ un subconjunto referido a una región de restricción global. Entonces una trayectoria de doblado relativa a R, es una trayectoria de doblado que pasa por completo dentro de la región R. La trayectoria de doblado óptima relativa a R, es la trayectoria de doblado que minimiza el costo entre todas las trayectorias relativas a R.

Dos regiones con restricción global son la banda de Sakoe-Chiba y el paralelogramo de Itakura mostradas en la Figura 2.6.

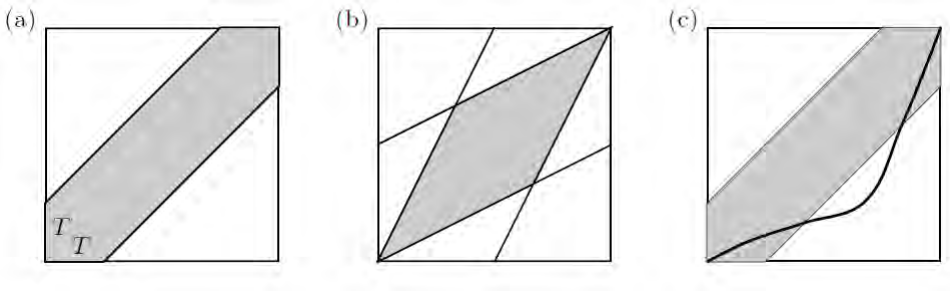

Figura 2.6: (a) Banda de Sakoe-Chiba de ancho T. (b) Paralelogramo de Itakura. (c) Trayectoria óptima de doblado que no pasa dentro de la región de restricción R [Muller07].

Los alineamientos de elementos pueden ser seleccionados únicamente de la región sombreada respectiva. Como ejemplo, considere la banda de Sakoe-Chiba que pasa a lo largo de la diagonal principal y que tiene un acho fijo T. Esta restricción implica que un elemento x_i puede ser alineado sólo a uno de los elementos y_j a favor de la diagonal multiplicando la distancia $d(x_i, y_j)$ de las celdas adyacentes por el peso asignado de acuerdo a la Figura 2.7[Itakura75, Sakoe78].

Así, la trayectoria óptima para ambas restricciones es encontrada por medio de las ecuaciones (2.23) y (2.24).

$$\delta_{Sakoe}(i,j) = min \begin{cases} \delta(i-1,j) + d(x_i, y_i) \\ \delta(i,j) + 2d(x_i, y_i) \\ \delta(i,j-1) + d(x_i, y_i) \end{cases} \tag{2.23}$$

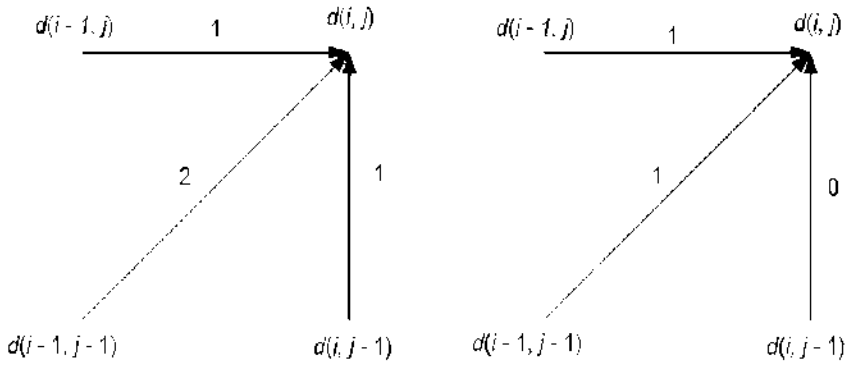

Figura 2.7: Restricciones globales; a la izquierda se muestra la restricción de Sakoe-Chiba y a la derecha la restricción de Itakura [Sakoe78, Itakura75].

$$\delta_{Itakura}(i,j) = \begin{cases} \delta(i-1,j) + d(x_i, y_i) \\ \delta(i,j) + d(x_i, y_i) \end{cases} \tag{2.24}$$

2.4. Modelos Estocásticos

En esta sección se presentan tres modelos estocásticos que son importantes para el desarrollo del trabajo. En principio se discute el método de estimación de ventanas de Parzen que será parte los enfoques utilizados para determinar la característica de *entropía por croma*. Posteriormente, se describe brevemente el algoritmo de mezcla de gaussianas que servirá como método de cuantización escalar más adelante. Finalmente, se presenta una breve revisión a los modelos ocultos de Markov discretos.

2.4.1. Ventanas de Parzen

Los métodos de estimación de la función de densidad de probabilidad confían en el hecho de que la probabilidad P de que un vector $x^{'}$ se encuentre en la región $\mathbb{R}$ esté dada por la ecuación(2.25).

$$P = \int_{\mathbb{R}} p\left(x^{'}\right) dx^{'} \tag{2.25}$$

Si ahora se asume que $p(x)$ es continua y que la región $\mathbb{R}$ es tan pequeña de tal manera que P no varíe apreciablemente dentro de ésta, la ecuación anterior se puede reescribir como se presenta en la ecuación (2.26),

$$P = \int_{\mathbb{R}} p\left(x^{'}\right) dx^{'} \cong p\left(x\right) V \tag{2.26}$$

donde x es un punto dentro de $\mathbb{R}$ y V es un volumen encerrado por $\mathbb{R}$. Por otro lado, suponga n muestras $x_1, ..., x_n$ independientes e idénticamente distribuidas de acuerdo a una función de probabilidad $p\left(x\right)$. Suponga que hay k de las n muestras fuera de la región $\mathbb{R}$, así, la probabilidad P de estas k muestras se puede aproximar a partir de la fracción media de muestras que caen en $\mathbb{R}$, de forma que P estará dada por la ecuación (2.27).

$$P = \frac{k}{n} \tag{2.27}$$

De esta manera, se llega al siguiente estimado para $p\left(x\right)$ (ecuación (2.28)) combinando las ecuaciones (2.26) y (2.27).

$$p\left(x\right) = \frac{k/n}{V} \tag{2.28}$$

Para estimar la densidad en x por medio de la ecuación (2.29), se forma una secuencia de regiones $\mathbb{R}_1, \mathbb{R}_2, ..., \mathbb{R}_n$ conteniendo a x. La primera región es usada con una muestra, la segunda con dos y así sucesivamente. Sea V_n el volumen de $\mathbb{R}_n$, k_n el número de muestras dentro de $\mathbb{R}_n$ y $p_n\left(x\right)$ el n-ésimo estimado para $p\left(x\right)$.

$$p_n\left(x\right) = \frac{k_n/n}{V_n} \tag{2.29}$$

El enfoque de ventanas de Parzen puede ser introducido asumiendo temporalmente que la región $\mathbb{R}_n$ es un hipercubo d-dimensional [Duda01]. Si h_n es la longitud de un lado de ese hipercubo, entonces su volumen está dado por la ecuación (2.30).

$$V_n = h_n^d \tag{2.30}$$

Definiendo la función ventana dada en la ecuación (2.31), se puede obtener una expresión analítica para el número de muestras k_n encerradas por el hipercubo,

$$\upsilon\left(u\right) = \begin{cases} 1, & |u_j| \leq \frac{1}{2} \quad j = 1, 2, ..., d \\ 0, & \text{por otro parte} \end{cases} \tag{2.31}$$

así, $\upsilon\left(u\right)$ define un hipercubo unitario centrado en el origen. Esto conduce a que $\upsilon\left(\frac{x - x_i}{h_n}\right)$ sea igual a la unidad si x_i cae dentro del hipercubo de volumen V_n centrado en x, y cero de otra manera. El número de muestras en este hipercubo por lo tanto está

dado por la ecuación (2.32).

$$k_n = \sum_{i=1}^{n} \upsilon \left(\frac{x - x_i}{h_n} \right) \tag{2.32}$$

Al sustituir la ecuación (2.32) en (2.29), se obtiene el estimado dado en la ecuación (2.33).

$$p_n(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{V_n} \upsilon \left(\frac{x - x_i}{h_n} \right) \tag{2.33}$$

Se puede generalizar la idea y usar otras funciones ventana con el propósito de tener otros métodos de estimación con ventanas de Parzen. Usando una distribución gaussiana como ventana, para el caso d-dimensional, el estimado de la densidad de probabilidad $p_n(x)$ está determinado por la ecuación (2.34), que es solo el promedio de n distribuciones gaussianas con media en el vector x_i. Ahora, la matriz de covarianzas Σ es un parámetro libre equivalente a las longitudes de cada lado del hipercubo.

$$p_n(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} e^{\frac{-(x - x_i)^T \Sigma^{-1} (x - x_i)}{2}} \tag{2.34}$$

2.4.2. Modelo de mezcla de gaussianas

Un Modelo de Mezcla de Gaussianas (MMG) es una función de densidad de probabilidad paramétrica que consiste de una combinación lineal de componentes gaussianos. Un MMG es comúnmente usado como un modelo paramétrico de la función de densidad de probabilidad de mediciones continuas o características de un sistema. Los parámetros de un MMG son estimados a partir de conjuntos de datos usando el algoritmo iterativo EM.

El algoritmo EM (por sus siglas en inglés de Expectation-Maximisation) es un método de estimación de máxima verosimilitud que se emplea para calcular los parámetros del modelo de la función de densidad de probabilidad de K-mezcla de gaussianas para un conjunto de datos dado. Para aplicar el algoritmo EM se necesita primero definir los conjuntos de datos completos e incompletos. Como siempre, los vectores de observación $[y_i,\ i = 1, ..., n-1]$ forman el conjunto de los datos incompletos. Los datos completos pueden ser vistos como los vectores de observación con una etiqueta k adjunta a cada vector y_i para indicar el componente del modelo de mezcla de gaussianas que generó el vector. Note que si cada vector y_i tiene adjuntada la etiqueta de un componente de la

mezcla, entonces el cálculo del vector de media μ y la matriz de covarianza Σ de cada componente de la mezcla sería relativamente simple. Por lo tanto, los datos completos e incompletos pueden ser definidos como sigue:

Los datos incompletos y_i, $i = 1, ..., n-1$.
Los datos completos $x_i = [y_i, k] = y_i(k)$, $i = 1, ..., n-1$ y $k \in (1, ..., K)$.

El método EM se enuncia como sigue: Dados n vectores de observación $y_1, y_2, ..., y_n \in \mathbb{R}^d$ junto con un MMG con K componentes, considerar el problema de estimar su conjunto de parámetros $\theta = \{w_j, \mu_j, \Sigma_j\}_{j=1}^{K}$, donde θ son los parámetros de la mezcla de funciones gaussianas como la que se presenta en la ecuación (2.35).

$$\phi(y|\mu, \Sigma) \triangleq \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} e^{\left(-\frac{1}{2}(y-\mu)^T \Sigma^{-1}(y-\mu)\right)} \tag{2.35}$$

Si se define $\varsigma_{ij}^{(m)}$ como la probabilidad de que en la m-ésima iteración, el i-ésimo vector pertenezca a la j-ésima componente gaussiana, esto es,

$$\varsigma_{ij}^{(m)} \triangleq p\left(Z_i = j | Y_i = y_i, \theta^{(m)}\right) = \frac{w_j^{(m)} \phi\left(y_i | \mu_j^{(m)}, \Sigma_j^{(m)}\right)}{\sum_{l=1}^{K} w_l^{(m)} \phi\left(y_i | \mu_l^{(m)}, \Sigma_l^{(m)}\right)} \tag{2.36}$$

y

$$n_j^{(m)} = \sum_{i=1}^{n} \varsigma_{ij}^{(m)} \tag{2.37}$$

cual satisface $\sum_{j=1}^{K} \varsigma_{ij}^{(m)} = 1$, entonces el método EM puede ser descrito en dos pasos, el paso E y el paso M. Así, el paso E queda descrito evaluando las ecuaciones (2.36) y (2.37), mientras que en el paso M se deben estimar los parámetros θ del modelo mediante las ecuaciones (2.38), (2.39) y (2.40) usando la restricción de que $\sum_{j=1}^{K} w_j = 1$.

$$w_j^{(m+1)} = \frac{n_j^{(m)}}{\sum_{j=1}^{K} n_j^{(m)}} = \frac{n_j^{(m)}}{n}, \quad j = 1, ..., K \tag{2.38}$$

$$\mu_j^{(m+1)} = \frac{1}{n_j^{(m)}} \sum_{i=n}^{n} \varsigma_{ij}^{(m)} y_i, \quad j = 1, ..., K \tag{2.39}$$

$$\Sigma_j^{(m+1)} = \frac{1}{n_j^{(m)}} \sum_{i=1}^{n} \varsigma_{ij}^{(m)} \left(y_i - \mu_j^{(m+1)} \right) \left(y_i - \mu_j^{(m+1)} \right)^T, \quad j = 1, ..., K \tag{2.40}$$

En la Tabla 2.1 se proporciona el algoritmo completo para estimar los parámetros de un MMG. Para ver a detalle cómo se determinan las ecuaciones de este algoritmo ver [Bilmes98, Chen10].

Tabla 2.1: Algoritmo EM para estimar los parámetros de un MMG.

1. **Inicialización.** Elegir los estimados iniciales para $w_j^{(0)}$, $\mu_j^{(0)}$ y $\Sigma_j^{(0)}$ para $j = 1, ..., K$ y calcular la log- verosimilitud inicial mediante
$L^{(0)} = \frac{1}{n} \sum_{i=1}^{n} log \left(\sum_{j=1}^{K} \omega_j^{(0)} \phi \left(y_i \vert \mu_j^{(0)}, \Sigma_j^{(0)} \right) \right)$
2. **Paso E.** Calcular
$\varsigma_{ij}^{(m)} = \frac{w_j^{(m)} \phi \left(y_i \vert \mu_j^{(m)}, \Sigma_j^{(m)} \right)}{\sum_{l=1}^{K} w_l^{(m)} \phi \left(y_i \vert \mu_l^{(m)}, \Sigma_l^{(m)} \right)}, \ i = 1, ..., n, \ j = 1, ..., K$ y $n_j^{(m)} = \sum_{i=1}^{n} \varsigma_{ij}^{(m)}, \ j = 1, ..., K$
3. **Paso M,** Calcular los nuevos estimados
$\omega_j^{(m+1)} = \frac{n_j^{(m)}}{n}, \ j = 1, ..., K,$ $\mu_j^{(m+1)} = \frac{1}{n_j^{(m)}} \sum_{i=1}^{n} \varsigma_{ij}^{(m)} y_i, \ j = 1, ..., K,$ $\Sigma_j^{(m+1)} = \frac{1}{n_j^{(m)}} \sum_{i=1}^{n} \varsigma_{ij}^{(m)} (y_i - \mu_j^{(m+1)})(y_i - \mu_j^{(m+1)})^T, \ j = 1, ..., K$
4. **Checar Convergencia.** Calcular la nueva log-verosimilitud
$L^{(m+1)} = \frac{1}{n} \sum_{i=1}^{n} log \left(\sum_{j=1}^{K} \omega_j^{(m+1)} \phi \left(y_i \vert \mu_j^{(m+1)}, \Sigma_j^{(m+1)} \right) \right)$ regresar al paso 2 si $\left\vert L^{(m+1)} - L^{(m)} \right\vert > \varepsilon$ si ε es un umbral. Por otro lado, terminar el algoritmo.

2.4.3. Modelos ocultos de Markov

Un Modelo Oculto de Markov (HMM por sus siglas en inglés de Hidden Markov Models) es un autómata de estados finitos capaz de producir a su salida una secuencia de símbolos observable. El autómata está formado por un conjunto de estados y evoluciona pasando de un estado a otro de forma probabilística. Los estados están conectados unos a otros por arcos de transición, con probabilidades asociadas a cada arco. Cada estado tiene asociada una función de densidad de probabilidad que define la probabilidad que tiene cada símbolo de ser emitido cada vez que se produce una transición desde dicho estado del HMM. Por lo tanto, un HMM consta de dos procesos estocásticos, la producción de símbolos y la secuencia de los estados en la evolución del mismo. De ellos, sólo la producción de símbolos es observable. Por este motivo se denomina a este autómata modelo oculto de Markov. La Figura 2.8 representa un HMM de tres estados con una posible secuencia de símbolos observables generada.

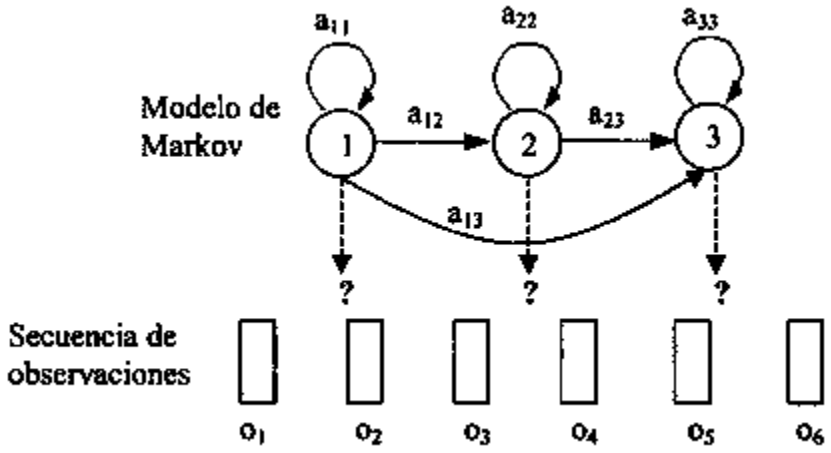

Figura 2.8: Esquema de producción de observaciones generadas por un HMM de tres estados.

Los elementos que constituyen un modelo HMM son cinco:

1. Un conjunto de N estados

$$S = \{s_1, s_2, ..., s_N\}$$

Los estados deben estar conectados entre sí, de forma que cualquiera de ellos pueda ser alcanzable desde al menos un estado.

2. Un conjunto de M símbolos observables que pueden ser producidos por el HMM

$$O = \{o_1, o_2, ..., o_M\}$$

3. Una matriz de probabilidades de transición de estados, $A = \{a_{ij}\}$. Esta matriz es cuadrada de dimensión N y cada elemento a_{ij} corresponde a la probabilidad de transición del estado s_i al s_j:

$$a_{ij} = p\left(q_t = s_j | q_{t-1} = s_i\right), \quad 1 \leq i, j \leq N$$

donde q_t indica el estado en el que se encuentra el modelo en el instante t. Debido a su naturaleza probabilística, cada elemento a_{ij} debe cumplir:

$$0 \leq a_{ij} \leq 1, \quad 1 \leq i, j \leq N$$

Por otra parte, también debe cumplirse que las probabilidades con origen en el mismo estado deben estar normalizadas:

$$\sum_{j=1}^{N} a_{ij} = 1$$

4. Un conjunto de parámetros $B = \{b_i\left(k\right),\ 1 \leq i \leq N,\ 1 \leq k \leq M\}$ que definen, para cada estado la función de densidad de probabilidad de producciones, si las observaciones son magnitudes continuas; o las distribuciones de probabilidad si las observaciones son discretas. Cada b_i se define de la siguiente forma:

$$b_i\left(o\right) = p\left(x_t = o | q_t = s_i\right), \quad 1 \leq i \leq N$$

donde x_t representa el valor de la observación en el instante de tiempo t. Se supone que la generación de observaciones depende sólo del estado en el que se encuentre el modelo en cada instante.

5. Un conjunto de probabilidades de estado inicial $\Pi = \{\pi_i\}$, siendo π_i la probabilidad de que el estado inicial del HMM sea el s_i:

$$\pi_i = p\left(q_1 = s_i\right), \quad 1 \leq i \leq N$$

Al igual que las probabilidades de transición de estados a_{ij}, las de estado inicial π_i deben verificar:

$$0 \leq \pi_i \leq 1, \quad 1 \leq i \leq N$$

$$\sum_{i=1}^{N} \pi_i = 1$$

De esta forma un HMM, queda definido completamente al especificar los conjuntos Π, A y B, que identificarán el modelo λ.

$$\lambda = (\Pi, A, B)$$

En el caso general, la variable estocástica asociada a la producción de observaciones es continua y multivariada. Sea X_1^T una secuencia observable, donde el subíndice 1 denota la observación en el instante de tiempo $t = 1$ y el superíndice T denota la observación en el instante de tiempo $t = T$, así, la secuencia está compuesta por T vectores continuos observados, de forma que $X_1^T = x_1, x_2, ..., x_T$.

La utilización de los HMM dentro de un sistema de reconocimiento, requiere la resolución de tres problemas:

- ***Evaluación***. Dada una secuencia de observaciones $X_1^T = x_1, x_2, ..., x_T$ y un modelo λ, se busca cómo evaluar la probabilidad $p\left(X_1^T|\lambda\right)$ de que la secuencia observada sea producida por dicho modelo.
- ***Estimación***. Dada una secuencia de observaciones $X_1^T = x_1, x_2, ..., x_T$ y un modelo λ, cómo elegir los parámetros del modelo $\lambda = (\Pi, A, B)$ para que la probabilidad de generación de dicha secuencia por el modelo sea óptima.
- ***Decodificación***. Dada una secuencia de observaciones $X_1^T = x_1, x_2, ..., x_T$ cómo obtener la secuencia de estados $Q_1^T = q_1, q_2, ..., q_T$ que mejor explica la generación de la secuencia por parte del modelo λ.

Los HMM se pueden utilizar para evaluar la probabilidad de que una secuencia sea producida por un modelo basándose en el proceso de producción de secuencias. Formalmente, conocida una secuencia de observaciones $X_1^T = x_1, x_2, ..., x_T$ y el correspondiente modelo de Markov $\lambda = (\Pi, A, B)$, para evaluar la probabilidad de que dicha secuencia sea producida por el modelo, $p\left(X_1^T|\lambda\right)$, se consideran todas las posibles secuencias de T estados. Si $Q_1^T = q_1, q_2, ..., q_T$ es una de dichas posibles secuencias de T estados, la probabilidad de la secuencia observada viene dada por la ecuación (2.41)

$$p\left(X_1^T|\lambda\right) = \sum_{O_i^T} \pi_{q_1} b_{q_1}\left(x_1\right) a_{q_1 q_2} b_{q_2}\left(x_2\right) \cdots a_{q_{T-1} q_T} b_{q_T}\left(x_T\right) \tag{2.41}$$

Dos algoritmos eficientes para evaluar la expresión anterior son los procedimientos llamados Hacia Adelante y Hacia Atrás (en la literatura conocidos como los algoritmos Forward y Backward) respectivamente.

Las probabilidades hacia adelante $\alpha_i(t)$ (ecuación (2.42)) se definen como la probabilidad de la secuencia parcial de observaciones compuesta por los símbolos presentados hasta el instante de tiempo t, cuando el estado en dicho instante t es s_i.

$$\alpha_i(t) \equiv p\left(x_1, x_2, ..., x_t, q_t = s_i|\lambda\right) \tag{2.42}$$

Es posible calcular $\alpha_i(t)$ de manera recursiva a partir de la inicialización, que es inmediata para $t = 1$, esto se presenta en la ecuación (2.43).

$$\alpha_i(1) = p\left(x_1, q_1 = s_i|\lambda\right) = \pi_i b_i\left(x_1\right), \quad 1 \leq i \leq N \tag{2.43}$$

Para los instantes sucesivos se puede expresar $\alpha_i(t)$ en función de los valores anteriores teniendo en cuenta las probabilidades de transición y de producción de símbolos (ecuación (2.44)).

$$\alpha_i(t) = \sum_{j=1}^{N} \alpha_j(t-1)\, a_{ji} b_i\left(x_t\right), \qquad \begin{matrix} 1 \leq t \leq T \\ 1 \leq i \leq N \end{matrix} \tag{2.44}$$

Finalmente, la probabilidad total de realizar una observación será la suma de las correspondientes a cada estado considerado como final, ésta se puede calcular por medio de la ecuación (2.45).

$$p\left(X_1^T|\lambda\right) = \sum_{i=1}^{N} \alpha_i(T) \tag{2.45}$$

De esta forma se reduce el número de operaciones mediante la eliminación de la dependencia exponencial con el número de símbolos de la secuencia a evaluar.

Las probabilidades hacia atrás $\beta_i(t)$ (ecuación (2.46)) se definen como la probabilidad de la secuencia de operaciones parcial constituida por todos los símbolos presentados a partir del instante de tiempo t, cuando el estado en dicho instante es s_i.

$$\beta_i(t) \equiv p(x_{t+1}x_{t+2}\cdots x_T, q_t = s_i|\lambda) \tag{2.46}$$

Si se inicializa a 1 para T, que es el valor conocido, entonces se tiene,

$$\beta_i(T) = 1, \quad 1 \leq i \leq N \tag{2.47}$$

De la misma forma que para las probabilidades hacia adelante, es posible calcular $\beta_i(t)$ de manera recursiva mediante la ecuación (2.48).

$$\beta_i(t) = \sum_{j=1}^{N} \beta_j(t+1)\, a_{ij} b_j(x_{t+1}), \qquad \begin{matrix} t = T-1, T-2, ..., 1 \\ 1 \leq i \leq N \end{matrix} \tag{2.48}$$

La decodificación de la secuencia de estados más probable, dada una secuencia $X_1^T = x_1, x_2, ..., x_T$ y un modelo $\lambda = (\Pi, A, B)$, no tiene única solución pues ésta depende del criterio utilizado para determinar la mejor secuencia. El criterio que se suele utilizar para escoger la secuencia óptima de estados $Q_1^{T^*}$ es maximizar la probabilidad condicionada de generación de la observación mediante la ecuación 2.49.

$$Q_1^{T^*} = \underset{Q_1^T}{\operatorname{argmax}}\, p(X_1^T|Q_1^T, \lambda) \tag{2.49}$$

La resolución de esta ecuación se hace utilizando el llamado algoritmo de Viterbi. Se trata de un algoritmo iterativo en el que se define una variable $\delta_i(t)$ como la probabilidad máxima de generación de una secuencia de t símbolos sobre cualquier secuencia simple de estados cuyo estado final es el s_i.

Para conocer la secuencia de estados, es necesario almacenar los valores del argumento que maximizan $\delta_i(t)$. Para ello se utiliza una matriz en la que cada elemento $\varphi_{it} = \varphi_i(t)$ contiene el índice del estado que maximiza la expresión anterior en el tiempo t. A partir de aquí se establece la condición inicial, la condición de terminación y como realizar las iteraciones:

1. Inicialización

$$\varphi_1(1) = 0$$

$$\delta_i(1) = \pi_i b_i(x_1), \quad 1 \leq i \leq N \tag{2.50}$$

2. Recursión

$$\delta_i(t) = \max_{1 \leq j \leq N} \delta_j(t-1) a_{ji} b_i(x_t), \quad 2 \leq t \leq T, \quad 1 \leq i \leq N \tag{2.51}$$

$$\varphi_i(t) = \operatorname*{argmax}_{1 \leq j \leq N} \delta_j(t-1) a_{ji}, \quad 2 \leq t \leq T, \quad 1 \leq i \leq N \tag{2.52}$$

3. Terminación

$$p^*(X_1^T|\lambda) = \max_{1 \leq i \leq N} \delta_i(T) \tag{2.53}$$

$$q_T^* = \operatorname*{argmax}_{1 \leq i \leq N} \delta_i(T) \tag{2.54}$$

4. Recursión para obtener la secuencia de estados

$$q_t^* = \varphi_{q_{t+1}^*}(t+1), \quad T-1 \geq t \geq 1 \tag{2.55}$$

El problema de entrenamiento implica, la estimación de los parámetros del modelo $\lambda = (\Pi, A, B)$, dada la secuencia de observación de entrenamiento $X_1^T = x_1, x_2, ..., x_T$, de tal forma que se maximice $p\left(X_1^T|\lambda\right)$. Este problema se trata evaluando el algoritmo de Baum Welch, aunque también existen otros métodos como el algoritmo de máxima verosimilitud (EM).

El algoritmo de Baum Welch garantiza la convergencia uniforme hacia un máximo local de la función probabilidad de generación. El algoritmo realiza en cada iteración una estimación del conjunto de parámetros y luego maximiza la probabilidad de generar los datos de entrenamiento utilizando el modelo, de tal modo que la nueva probabilidad es mayor o igual a la previa. El procedimiento de entrenamiento define una función de probabilidad a posteriori $\gamma_i(t)$ (ecuación (2.56)), la probabilidad de estar en el estado

i en el instante t, dada la secuencia de observación X_1^T y el modelo λ.

$$\gamma_i(t) = p\left(q_t = s_i | X_1^T, \lambda\right)$$

$$\gamma_i(t) = \frac{\alpha_i(t)\,\beta_i(t)}{\sum_{i=1}^{N} \alpha_i(t)\,\beta_i(t)} \tag{2.56}$$

Definiendo otra función de probabilidad $\xi_{ij}(t)$, la probabilidad de estar en el estado i en el instante t e ir al estado j en el instante $t+1$, dado el modelo λ y la secuencia de observación X_1^T. La función $\xi_{ij}(t)$ puede ser matemáticamente definida mediante la ecuación (2.57).

$$\xi_{ij}(t) = p\left(q_t = s_i, q_{t+1} = s_j | X_1^T, \lambda\right)$$

$$\xi_{ij}(t) = \frac{\alpha_i(t)\,a_{ij} b_j(x_{t+1})\,\beta_j(t+1)}{\sum_{i=1}^{N} \alpha_i(t)\,\beta_i(t)} \tag{2.57}$$

La relación entre $\gamma_i(t)$ y $\xi_{ij}(t)$ esta definida mediante la ecuación (2.58).

$$\gamma_i(t) = \sum_{j=1}^{N} \xi_{ij}(t) \tag{2.58}$$

Ahora, si $\gamma_i(t)$ es sumada sobre todos los instantes (excluyendo el instante T) se obtiene el número esperado de veces que el estado s_i ha sido visitado sobre todos los instantes. Por otro lado, la suma $\xi_{ij}(t)$ sobre todos los instantes (excluyendo el instante T) da el número esperado de transiciones que son hechas de i a j.

A partir de las definiciones para $\gamma_i(t)$ y $\xi_{ij}(t)$ con las ecuaciones (2.59), (2.60) y (2.61) se deducen los parámetros del modelo:

$$\pi_i^{'} = \gamma_i(1) = \frac{\alpha_i(1)\,\beta_i(1)}{\sum_{j=1}^{N} \alpha_j(1)\,\beta_j(1)}, \quad 1 \le i \le N \tag{2.59}$$

$$a_{ij}^{'} = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)} = \frac{\sum_{t=1}^{T-1} \alpha_i(t)\, a_{ij} b_j(x_{t+1})\, \beta_j(t+1)}{\sum_{t=1}^{T-1} \alpha_i(t)\, \beta_i(t)}, \quad \begin{matrix} 1 \leq i \leq N \\ 1 \leq j \leq N \end{matrix} \tag{2.60}$$

$$b_j^{'}(k) = \frac{\sum_{\substack{t=1 \\ x_t = O_k}}^{T-1} \gamma_j(t)}{\sum_{t=1}^{T-1} \gamma_j(t)}, \quad \begin{matrix} 1 \leq k \leq M \\ 1 \leq j \leq N \end{matrix} \tag{2.61}$$

Para una revisión más detallada sobre este tema ver [Rabiner89, Salcedo07, Trebbe95].

2.5. Comentarios

Hasta este punto el lector debe ser capaz de comprender los fundamentos teóricos sobre los cuales está basado este trabajo de investigación. En los próximos capítulos se asume que el lector está familiarizado con la terminología y que comprende todos los conceptos presentados en este capítulo.

Capítulo 3

ENTROPÍA POR CROMA

En este capítulo se discute la aportación que tiene este trabajo en el estado del arte respecto al tema de caracterización robusta de señales de audio. Este capítulo se encuentra dividido en cuatro secciones principales. La Sección 3.1 ayuda al lector a comprender la importancia que tiene el uso de la *entropía por croma* en el tópico de identificación de audio. Además, describe brevemente el tema sobre huellas de audio aplicado al problema de reconocimiento de audio. La Sección 3.2 presenta al lector el proceso para determinar la característica de audio basada en la *entropía por croma*. La Sección 3.3 está destinada para que el lector conozca los experimentos y resultados que son obtenidos al evaluar esta característica en un sistema de identificación de audio. Por último, la Sección 3.4 presenta al lector las conclusiones y comentarios de este capítulo.

3.1. Introducción

En este capítulo se presenta una nueva característica de audio que sirve para diferentes propósitos en tareas de reconocimiento de audio. A esta característica se le dio el nombre de *Entropía por Croma* (EC) y se puede definir como una medida del nivel de contenido de información que tiene una señal de audio por *croma*. Se pueden enumerar dos cuestiones importantes para la EC, ¿Qué tan inmune es respecto al ruido? y ¿Cuál es su interpretación en la música? Para tratar de responder a la primera pregunta haremos referencia al trabajo presentado en [Camarena09] donde se introduce una huella de audio que tiene una elevada inmunidad al ruido. El proceso de extracción de la característica que utiliza esta huella se basa en extraer la *entropía espectral multibanda* (Sección 2.2.2). Esta huella de audio logra excelentes resultados para reconocer señales ante diferentes clases de degradación. Si tomamos por hecho de que por el uso de la *entropía* la huella de audio anterior logra esa elevada inmunidad al ruido, entonces

podemos afirmar que la EC también es muy robusta en ese sentido. Con respecto a la segunda pregunta, podemos mencionar que la EC fue diseñada para tener una similar interpretación en la música a la que tienen los *cromagramas* (Sección 2.2.1) en el sentido de mostrar el contenido armónico y melódico por croma de una señal de audio. En conclusión se buscó que la EC fuera robusta al ruido y a las variaciones dinámicas de una señal de audio.

Otro aspecto importante que se tiene que mencionar sobre la EC es que se trata de una característica diseñada específicamente para usarse en el problema de alineamiento de señales en tiempo real en aplicaciones de seguimiento de audio. Esto es importante que quede claro ya que la característica a la que se hace más referencia en la literatura para este problema son los *cromagramas* y es por esta razón que únicamente comparamos contra éstos en los experimentos presentados en este capítulo. Básicamente el objetivo de este capítulo es probar que la EC es más robusta que los *cromagramas* para degradaciones típicas de un escenario donde se lleva a cabo una presentación en vivo. La manera en cómo se prueba lo anterior es utilizando el enfoque de huellas de audio (las pruebas con la EC sobre variaciones dinámicas en tiempo y amplitud de una señal se exponen en el Capítulo 5). A continuación se proporciona una breve introducción del tema de HA.

Los sistemas basados en Huellas de Audio (HA) son mejor conocidos por su aplicación de enlazar audio sin etiquetar o desconocido a sus correspondientes metadatos (ej. intérprete, nombre de la pieza musical, etc.). Una HA es usada para propósitos de identificación y monitoreo, por lo tanto, la huella debe ser única, compacta y robusta como sea posible a degradaciones comunes en las señales de audio [Cano05b]. En una HA el proceso de extracción de características normalmente está determinado sobre una base de extracción de tramas de audio, esto es, cada determinado tiempo una trama o segmento de audio es extraído de la señal para ser analizado.

A pesar de que existen diferentes enfoques para la identificación de HA, todos ellos siguen dos procesos fundamentales: La extracción de la huella y el algoritmo de reconocimiento. El proceso de extracción de la huella está dividido en un bloque de extracción y un bloque de modelado. El bloque de extracción determina un conjunto de medidas y características a partir de la señal cumpliendo con las siguientes peculiaridades: a) reducir la dimensionalidad, b) extraer parámetros perceptualmente significativos, c) poseer invarianza o robustez y d) tener una correlación temporal. Por lo tanto, la finalidad de este bloque es realizar el procesamiento de la señal de audio para derivar un conjunto de características perceptuales de una manera concisa y robusta. Hay varias técnicas para extraer estas características perceptuales y se pueden determinar ya sea

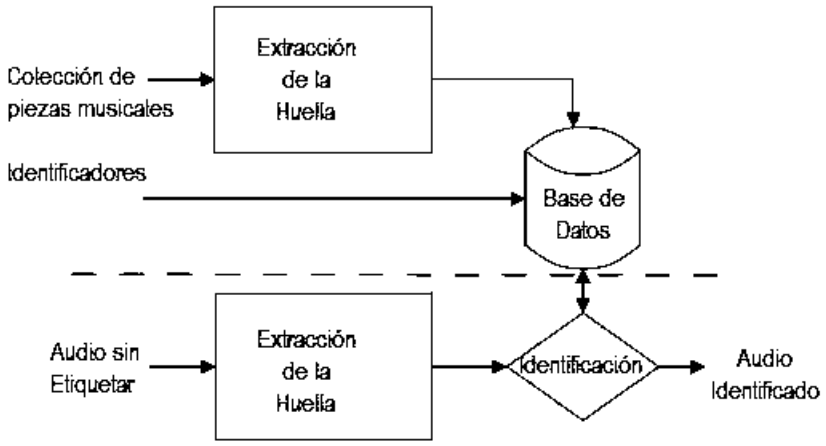

Figura 3.1: Esquema de identificación de audio basado en huellas de audio [Cano05a].

en el dominio del tiempo o de la frecuencia a través de una variedad de transformaciones lineales, tales como: Transformada Discreta de Fourier (TDF), Transformada Discreta Coseno (TDC), Transformada de Haar, Transformada de Walsh-Hadamard, Transformada Wavelet, entre otras.

El bloque de modelado define la caracterización final de la HA (ej. un vector o trazo de vectores de números reales, un libro de códigos, una secuencia de índices, palabras binarias o atributos de alto nivel musicalmente significantes). Una HA debe incluir los siguientes requerimientos: a) Potencia de discriminación, b) Invarianza a distorsiones, c) Compacidad y d) Simplicidad Computacional.

Los requerimientos mencionados implican un compromiso entre reducción de la dimensionalidad y pérdida de información. Una vez determinada la huella, se usa un algoritmo de reconocimiento para buscar esa huella dentro de una base de datos de HA. Este algoritmo debe encontrar el elemento de la base de datos que posea el segmento más semejante a la huella. En los algoritmos de reconocimiento es común usar distancias métricas para esta tarea.

Con respecto a los métodos de búsqueda, éstos deben ser rápidos, ya que el número de comparaciones entre huellas es grande y la determinación de la distancia métrica puede ser computacionalmente costosa. Una buena alternativa es usar métodos de indexamiento [Sadit12]. En la Figura 3.1 se muestra el esquema general propuesto en [Cano05a] de un sistema de reconocimiento de HA. En la siguiente sección se presenta el proceso para determinar la EC y los *cromagramas de entropía* que sirven para mostrar la variación del contenido armónico y melódico por *croma* de una señal de audio.

3.2. Cromagramas de Entropía

El proceso de extracción de la característica de audio que se propone en esta sección utiliza la *entropía* de una variable aleatoria para estimar el nivel de contenido de información que tiene una señal en cada *croma*. Una manera de calcular la *entropía* es mediante el estimado de la función de densidad de probabilidad $p(x)$ de una señal. Para este propósito se pueden usar métodos paramétricos, no paramétricos e histogramas. En los métodos paramétricos es necesario tener un conocimiento a priori de cómo están distribuidos los datos. Distribuciones tales como la gaussiana, laplaciana, entre otras, se pueden usar para este propósito. Una vez que el tipo de distribución se elige, sus parámetros tienen que ser determinados. En los métodos no paramétricos no se asume nada acerca de la forma que toma la función de distribución de probabilidad. La función es determinada por los datos y por algún kernel sobre un proceso iterativo. El método de ventanas de Parzen y los K-vecinos más cercanos son algunos ejemplos de estos métodos no paramétricos [Duda01]. Finalmente, los histogramas son usados para agrupar datos en forma de barras, donde la superficie de cada barra es proporcional a la frecuencia de los datos representados. Se usan sobre todo, para obtener un panorama general de la función de distribución de probabilidad. En este trabajo usamos dos enfoques sobre la función de densidad de probabilidad; el primero asume una distribución normal con media de cero; el segundo utiliza un estimado de $p(x)$ mediante el método de ventanas de Parzen.

La EC consiste en extraer por cada trama de audio ciertos valores llamados *valores de entropía por croma* (VEC). Para determinar estos valores se consideran algunos de los parámetros descritos en la Sección 2.2.1 para *cromagramas*. El procedimiento comienza especificando una frecuencia inferior, f_0, la frecuencia de muestreo, f_s, y el número de componentes espectrales por cada octava, b. Mediante la ecuación 2.4 se hace $k = 0$ para encontrar el tamaño de ventana con mayor longitud, $N(0)$. Esta longitud de ventana se usa para determinar la cantidad de datos de audio a ser extraídos de la señal por cada trama de audio. Una trama de audio está compuesta por $N(0)$ datos de audio, a los cuales se les aplica una ventana de Hann, y por $N(0)$ ceros de relleno. La Figura 3.2 muestra un ejemplo de una trama de audio donde se observa claramente la distribución a la izquierda y a la derecha de los ceros que se anexaron a la trama.

Para obtener el espectro de una trama de audio se utiliza la TDF, con esto se asegura que $N(0)$ componentes espectrales estarán disponibles para el estimado de $p(x)$. No se considera la TQC porque el espectro Q constante posee pocos componentes espectrales para hacer un buen estimado de $p(x)$ en cada *croma*. El ejemplo siguiente

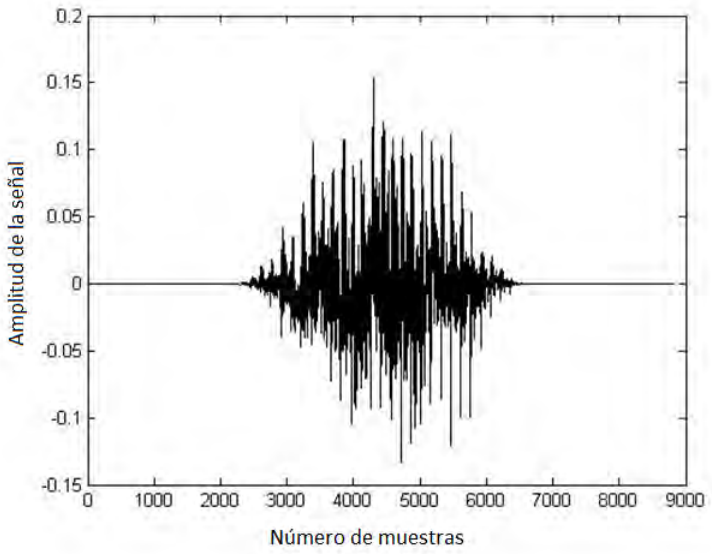

Figura 3.2: Ejemplo de una trama de audio con $N(0)$ datos de audio y $N(0)$ ceros de relleno.

demuestra esta afirmación. Considere $f_0 = 100\text{Hz}$, $f_{max} = 10\text{kHz}$, $f_s = 44.1\text{kHz}$ y $b = 12$ componentes por octava; la longitud de ventana tomando en cuenta estos parámetros es de $N(0) \approx 7416$ muestras, y el número de componentes espectrales sobre el espectro Q constante es de $K \approx 80$. De esta manera, por cada *croma* se tendrán a lo mucho $K/b \approx 7$ componentes, lo cual resulta insuficiente para hacer una buena estimación de $p(x)$. Ahora, si se considera la TDF sobre el ejemplo anterior, el número de componentes espectrales a tomar en cuenta es,

$$K = \frac{N(0)}{f_s}(f_{max} - f_0) \approx \frac{7416}{44100}(10000 - 100) \approx 1665$$

Tal cantidad de componentes permite hacer un mejor estimado de $p(x)$, ya que si se hace la correspondencia entre el espectro Q constante y el espectro de Fourier, se tendrían alrededor de 130 componentes por cada estimado de $p(x)$ en cada *croma*. En las Figuras 3.3 y 3.4 se muestran los espectros obtenidos de una trama de audio usando la TQC y la TDF, respectivamente, tomando en cuenta los parámetros del ejemplo. Es claro observar en estas figuras como el espectro obtenido mediante la TQC tiene menos resolución en frecuencia (menos coeficientes espectrales para representar el contenido espectral de la trama de audio) que el espectro obtenido por la TDF.

Cuando se usa la TQC para determinar los VC se divide el espectro Q constante en octavas, donde cada octava tiene b componentes espectrales a lo más. Así, los VC se obterminan como sigue: El primer *valor de croma* es obtenido sumando el primer

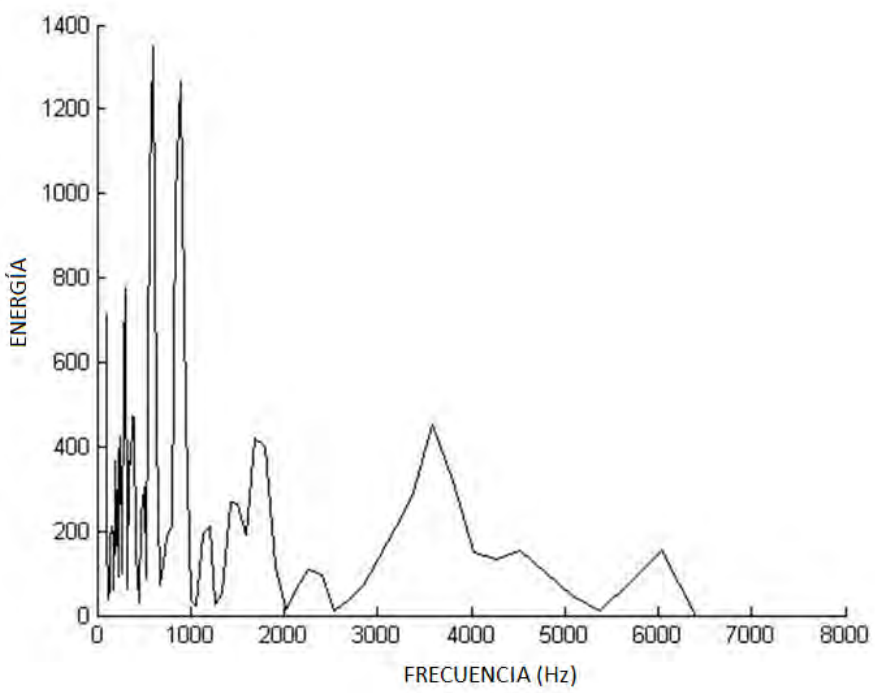

Figura 3.3: Espectro de una trama de audio obtenido por la TQC.

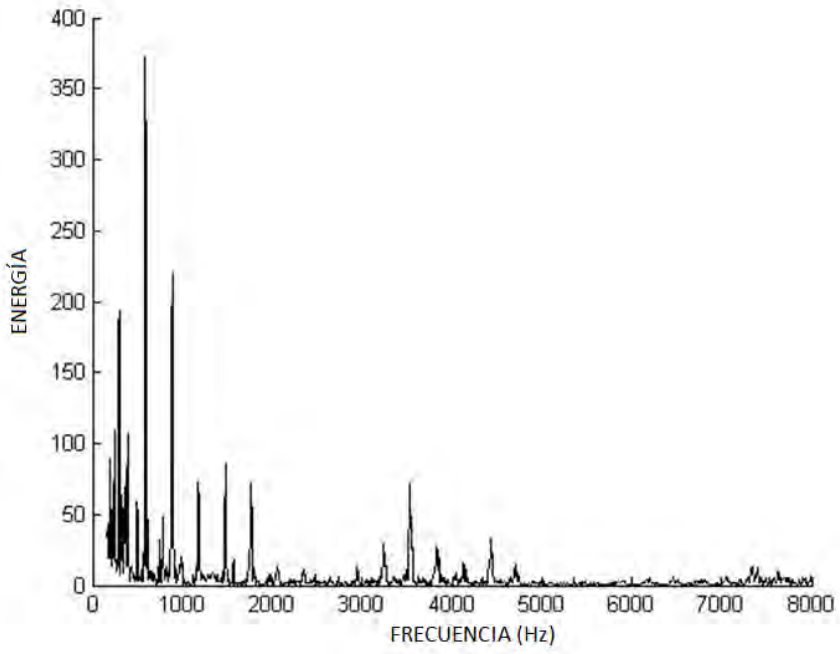

Figura 3.4: Espectro de una trama de audio obtenido por la TDF.

componente espectral de cada octava. El segundo *valor de croma* es obtenido sumando el segundo componente espectral de cada octava y así, sucesivamente.

Por otra parte, para determinar los VEC se considera el siguiente procedimiento: Dada una trama de audio, se aplica la TDF para obtener los componentes espectrales de la señal. A continuación, el primer *valor de entropía por croma* es obtenido determinando la *entropía* de los componentes espectrales que están delimitados en cada octava, por la frecuencia de los componentes usados para obtener el primero y segundo *valor de croma*. Para conseguir el segundo *valor de entropía por croma*, se determina la *entropía* considerando los componentes espectrales que son delimitados en cada octava, por la frecuencia de los componentes usados para obtener el segundo y tercero de los *valores de croma*. El resto de los *valores de entropía por croma* siguen un procedimiento similar.

Matemáticamente, el procedimiento anterior se puede expresar de la siguiente manera: Dada una frecuencia inferior, f_0, la frecuencia de muestreo, f_s, y el número de *valores de entropía por croma*, b, determinar la cantidad de datos, N, a ser extraídos de la señal de audio mediante la ecuación (3.1).

$$N = \frac{f_s}{f_0 \left(2^{1/b} - 1\right)} \tag{3.1}$$

Aplicar a los datos extraídos, $x\left(k\right)$, la ventana de Hann definida por la ecuación (3.2),

$$w\left(k\right) = \frac{1}{2} - \frac{1}{2} cos\left(\frac{2\pi k}{N-1}\right) \tag{3.2}$$

para $k = 0, 1, ..., N-1$. Para $k \notin 0, ..., N-1$ se tiene que $x\left(k\right) w\left(k\right) = 0$, por lo tanto, una trama de audio, $y\left(n\right)$, se puede determinar mediante la ecuación (3.3).

$$y\left(n\right) = x\left(n + \frac{N}{2}\right) w\left(n + \frac{N}{2}\right) \quad \forall\, n = -N, ..., N-1 \tag{3.3}$$

Para un tamaño de $2N$ de una trama de audio, la TDF se determina usando la ecuación (3.4),

$$X\left(l\right) = \sum_{n=0}^{2N-1} y\left(n\right) e^{\frac{-j2\pi nl}{2N}} \tag{3.4}$$

para $l = 0, 1, ..., 2N-1$. Los componentes espectrales considerados en cada *croma* deben ser elegidos de acuerdo a la siguiente relación de frecuencias,

$$f\left(i+md\right) \leq \frac{lf_s}{2N} \leq f\left(i+1+md\right)$$

para $i = 0, 1, ..., b-1$ y $m = 0, 1, ..., M$, donde $M = K/b$ y $d = b$. El término $\frac{lf_s}{2N}$ es la frecuencia de los componentes espectrales de $X\left(l\right)$ y $f\left(\cdot\right)$ es la frecuencia de los componentes espectrales del espectro Q constante. Para determinar la *entropía* de cada *croma* se consideran dos enfoques; el primero utiliza un estimado de la función de densidad de probabilidad, $p\left(x\right)$, por medio de la ecuación (2.34). Para este estimado se usa el módulo de los componentes espectrales en cada *croma*. De esta manera, la *entropía* se determina mediante la ecuación (2.7). En el segundo enfoque se asume que la parte real e imaginaria de los componentes espectrales se comportan como dos variables aleatorias independientes obedeciendo una distribución normal con media de cero, por lo tanto, se puede usar la ecuación (2.14) para determinar la *entropía*. Así, para este último enfoque los *valores de entropía por croma*, $VEC\left(d\right)$, se obtienen mediante la ecuación (3.5),

$$VEC\left(d\right) = H_d = ln\left(2\pi e\right) + \frac{1}{2} ln\left(\sigma_x^2 \sigma_y^2 - \sigma_{xy}^2\right) \tag{3.5}$$

donde, H_d, denota la *entropía* de los componentes espectrales del d-ésimo *croma*, σ_x^2 y σ_y^2, son las varianzas de la parte real e imaginaria respectivamente, y σ_{xy}^2, es la covarianza entre la parte real y la imaginaria del espectro en su forma rectangular.

Por cada trama de audio un vector con d *valores de entropía por croma* es obtenido. Un *cromagrama de entropía* está formado por una secuencia de estos vectores, de esta manera, se tiene una matriz con d renglones y un número de columnas que depende del número de tramas de audio consideradas. En la Figura 3.5 se muestra el *cromagrama de entropía* del mismo segmento de audio utilizado para generar la Figura 2.1. Comparando ambas figuras se puede observar que existe una gran similitud entre ellas en la forma en como varía el contenido armónico y melódico (regiones más oscuras) de la señal de audio. Así, un *cromagrama de entropía* intenta mostrar al igual que los *cromagramas*, el cambio armónico y melódico de una señal de audio. En la literatura a las regiones más oscuras de un *cromagrama* se les conoce como eventos de notas y representan el nivel de energía que tienen distintos tonos (notas musicales) en todas las octavas en un instante de tiempo. En un *cromagrama de entropía*, estas regiones se pueden considerar las zonas donde se tiene el mayor nivel de contenido de información atribuido a diferentes tonos en un instante de tiempo.

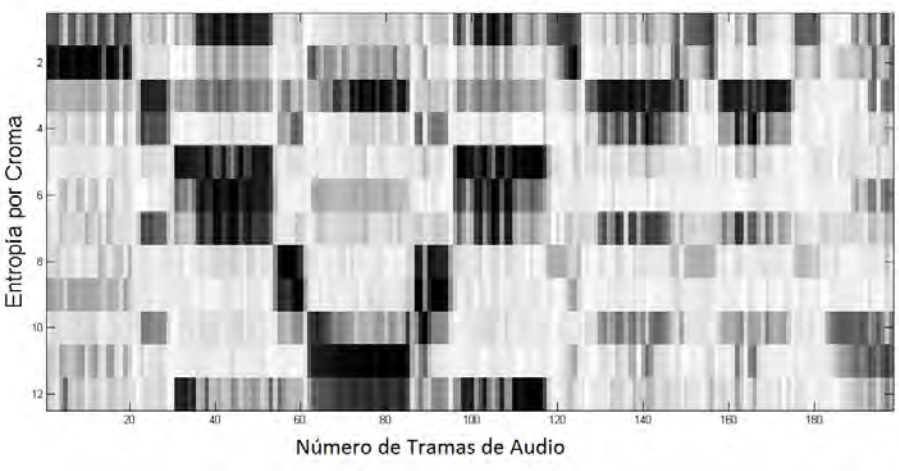

Figura 3.5: *Cromagrama de entropía* de un segmento de audio de 10 seg. de música monofónica.

3.3. Experimentos y Resultados

En esta sección se proporcionan los experimentos y resultados que son obtenidos al evaluar los *cromagramas de entropía* en un sistema de identificación de audio usando el enfoque de HA. El objetivo de estos experimentos es determinar el grado de inmunidad a ruido que logra la EC con respecto a la característica de audio basada en *cromagramas*. Para comprender en que consisten estos experimentos considere el siguiente problema:

Dado un conjunto de piezas musicales de diferentes tipos y géneros, se busca identificar si un pequeño trozo de una señal de audio pertenece a alguna de las piezas del conjunto. Este problema es sencillo si las piezas que se comparan no presentan ningún nivel de degradación o distorsión en la señal de audio. Sin embargo, este panorama sólo se presenta en ciertas situaciones donde el caso de estudio es controlado. Por lo tanto, es común encontrarnos con piezas musicales que son procesadas o alteradas por los diferentes medios a los cuales pueden estar expuestas. Así, este problema se vuelve interesante y tiende a ser complejo a medida que las piezas musicales se encuentren ecualizadas, tengan pérdidas de información por compresión o transmisión, presenten contaminación por ruido, sean regrababas, posean efectos de eco y reverberación o que estén interpretadas por otro intérprete y ejecutadas con diferentes instrumentos musicales.

Como se explicó anteriormente, la EC es una nueva característica de audio que permite mostrar la variación del contenido armónico y melódico de una señal, tal como hacen los *cromagramas*. Sin embargo, si comparamos los *cromagramas de entropía* contra *cromagramas* no se puede asumir nada acerca de cuál característica es más robusta

en presencia de ruido. Para esto se propone utilizar un sistema de identificación de audio basado en HA. Tal sistema permitirá probar cuantitativamente cual característica es más robusta a ruido.

3.3.1. Extracción de huellas de audio

Para la característica basada en *cromagramas* se propone una HA a la que llamaremos *huella de audio de valores de croma* o HAVC por sus iniciales. Para la característica basada en *cromagramas de entropía* se proponen dos tipos de HA. Para reconocer a una de la otra, se nombra como HAEC1 a la huella que utiliza *cromagramas de entropía* determinados a partir del cálculo de la *entropía* con el estimado de $p(x)$ usando la ecuación (2.34); la huella que utiliza *cromagramas de entropía* determinados a partir del cálculo de la *entropía* con la distribución normal con media de cero se le asigna el nombre de HAEC2. Las iniciales HAEC que refieren a las dos HA anteriores son por las iniciales de *huella de audio de entropía por croma*.

Con el enfoque de HA se busca tener una caracterización más compacta de estas características sin presentar cambios significativos. Estas tres HA siguen un procedimiento similar para determinarlas. Las consideraciones comunes a todas ellas son:

- Convertir la señal de audio a una señal monoaural $x(n)$ promediando ambos canales, si ésta es de tipo estereofónica.
- Aplicar a la señal $x(n)$ el filtro de pre-énfasis definido por $h(n) = x(n) - ax(n-1)$ para resaltar la energía en altas frecuencias. Este proceso ayuda a identificar mejor los eventos de nota. Se usa un factor de $a = 0.9$. Aquí, $x(n)$ es la señal de audio completa.
- Considerar tramas de audio de 100ms de duración (es decir, 4410 muestras para una frecuencia de muestreo de 44.1KHz). Para esta cantidad de muestras se tiene que considerar una frecuencia inferior, f_0, de 168.1715Hz y $b = 12$. En la literatura sobre HA se recomienda utilizar tramas de audio de duración entre 10 y 500ms para abarcar pocos eventos perceptuales en una trama [Haitsma02, Allamanche01, Wang03, Özer05, Venkatachalam04, Mottio08].
- Usar un traslape del 50 % entre cada trama. Esta cantidad de traslape permite tener 20 tramas por segundo de audio.
- Considerar una frecuencia superior, f_{max} de 8000Hz. Dado que la sensibilidad

auditiva de una persona adulta se encuentra en el rango de 200 a 8000Hz aproximadamente, no se consideran frecuencias mayores a 8000Hz.

Una vez extraídos los *cromagramas* y *cromagramas de entropía* de la señal, se codifican con la finalidad de tener una representación más compacta de los valores que los definen. El proceso para formar la HA consiste en codificar en un solo bit el signo de las diferencias entre valores consecutivos. Si el signo de estas diferencias es mayor o igual a cero, el bit toma el valor de 1 y en caso contrario tomará el valor de 0. Este proceso de codificación puede determinarse matemáticamente mediante la ecuación (3.6),

$$F(d,m) = \begin{cases} 1, & si\ [V(d,m) - V(d,m-1)] \geq 0 \\ 0, & por\ otro\ lado \end{cases} \tag{3.6}$$

donde $V(d,m)$ denota el d-ésimo valor obtenido de la trama de audio actual, m y $V(d,m-1)$ denota el d-ésimo valor obtenido de la trama de audio anterior, $m-1$. Para los experimentos se extraen 12 valores (debido a las 12 notas musicales) por cada trama de audio, por lo tanto, únicamente se necesitan dos bytes para codificar cada trama.

Una primera prueba de robustez para ambos tipos de huellas es presentada en la Figura 3.6. Para esta prueba se consideró un segmento de 2 segundos de audio monofónico y una versión degradada de éste. El segmento que contiene la versión degradada del audio se obtuvo mediante la adición de ruido blanco a la señal. La relación señal a ruido que se logró fue de 5dB. La prueba consistió en determinar HAVC y HAEC de ambos segmentos para tomar la diferencia en bits entre huellas. En la parte derecha de la Figura 3.6 se muestra en la parte superior la HAVC del segmento de audio original, en la parte central se muestra la HAVC del segmento de audio degradado y en la parte inferior se muestra una imagen binaria con los bits que son diferentes entre ambas huellas (el color blanco indica que los bits de ambas huellas permanecieron sin cambio). Para la HAVC, se tuvo una diferencia de 187 bits entre ambas huellas de un total de 444 bits, lo que indica que hay una variación entre las dos huellas del 42 % con este nivel de degradación. La parte izquierda de la Figura 3.6 corresponde a la prueba con HAEC. Siguiendo el mismo procedimiento anterior, para esta huella se obtuvo una diferencia de 150 bits del total, lo que indica que ambas huellas difieren en un 33 %. Este resultado preliminar es un indicativo que lleva a pensar que los *cromagramas de entropía* son más robustos que los *cromagramas*, cuando las señales de audio son distorsionadas por algún tipo de degradación.

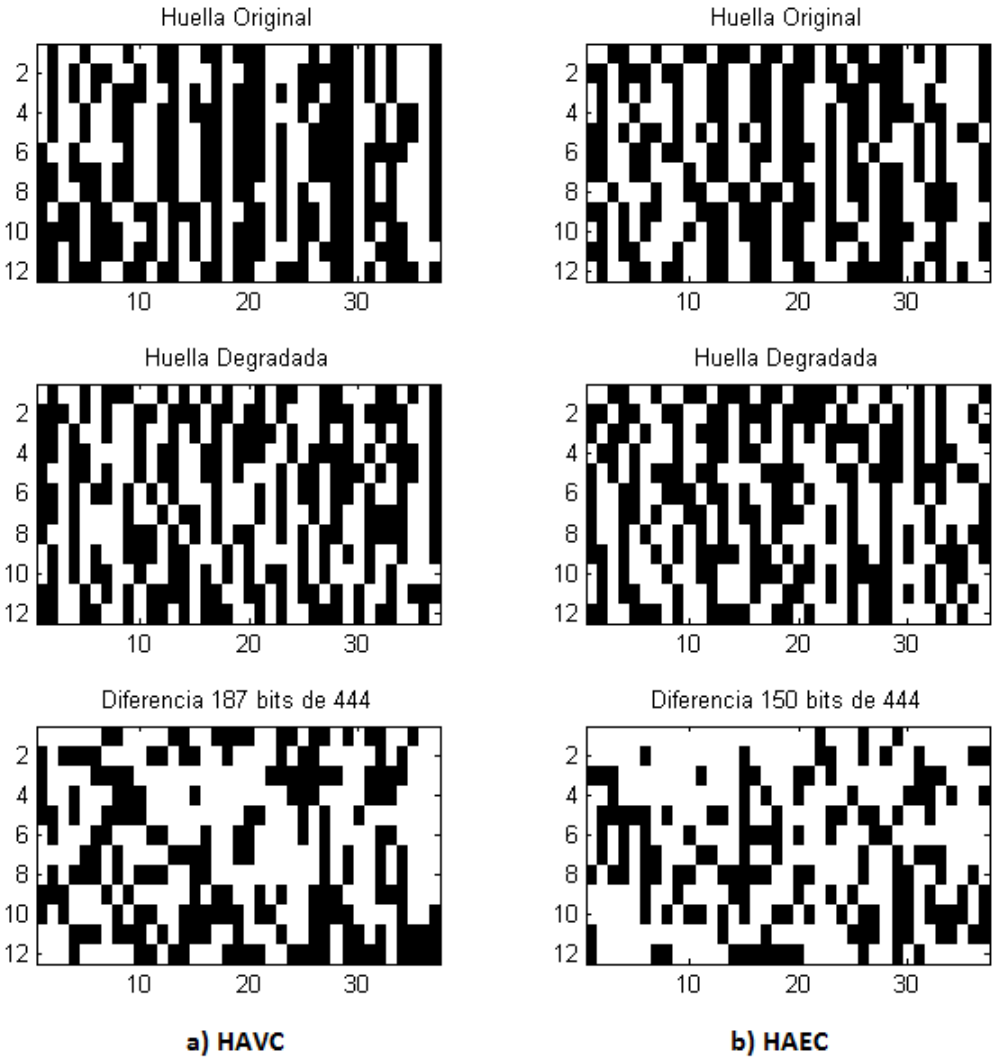

Figura 3.6: Prueba de conteo de bits diferentes entre HAVC y HAEC.

3.3.2. Base de datos

Los experimentos presentados en esta sección son realizados usando música monofónica y polifónica. La base de datos de la cual se obtuvo esta música, está compuesta por dos conjuntos; un conjunto de 300 piezas de música monofónica y un conjunto de 3500 piezas de música polifónica. El conjunto de música monofónica está constituido por piezas tocadas ya sea por flauta o piano en una sola línea melódica sin acompañamiento. El género de la música de este conjunto es en su mayoría de tipo instrumental y puede descargarse directamente de los siguientes enlaces, www.flutetunes.com y www.1001pianos.com. Por otro lado, el conjunto de música polifónica está constituido por piezas de todo tipo de género, entre los cuales destacan el rock, pop, balada, instrumental y regional. Cabe hacer notar que esta colección de canciones no se encuentra disponible en ningún servidor ya que se trata de una colección personal, sin embargo, ninguna infracción fue hecha.

3.3.3. Degradaciones

Los experimentos sobre robustez implican distorsionar las señales de audio mediante diferentes clases de degradaciones. Los tres tipos de degradación considerados son: a) Efecto de reverberación, b) Adición de ruido ambiental, y c) Transmisión por bocina a micrófono. Estas clases de degradaciones son típicas de un escenario donde un concierto se está llevando a cabo y los ecos del auditorio o teatro introducen un nivel de reverberación a la señal capturada, además del ruido que se suma de la gente que se encuentra gritando y aplaudiendo. La degradación por transmisión de bocina a micrófono intenta simular la grabación en tiempo real del evento en vivo.

La distorsión de las señales de audio con estos tres tipos de degradación fue llevada a cabo de la siguiente manera: La señal de audio original se distorsiona con el efecto de reverberación utilizando los parámetros dados en la Tabla 3.1. Después, para contaminar la señal con ruido ambiental, se mezcla la señal resultante del paso anterior con un segmento de audio grabado con el ruido de fondo de un evento en vivo, el cual consiste de gente que está gritando y aplaudiendo. Finalmente, para agregar a la señal la degradación por transmisión de bocina a micrófono, se reproduce la señal distorsionada del paso anterior por medio de una computadora y se recaptura por medio del micrófono de otra computadora. Esta clase de degradación involucra un filtrado pasa banda con frecuencias de corte igual al ancho de banda del micrófono y una atenuación por el medio de transmisión.

La relación señal a ruido (SNR, por sus siglas en inglés de Signal to Noise Ratio)

Tabla 3.1: Parámetros para reverberación.

Tipo	Auditorio
Longitud de Reverberación Total	1290ms
Tiempo de Ataque	52ms
Tiempo de Absorsión de Alta Frecuencia	721ms
Percepción	60 %
Mezcla de la Señal Original	95 %
Mezcla de Reverberación	135 %

alcanzada está en el rango de 2 a 3 decibeles. El SNR está definido por la ecuación (3.7), donde $P_{señal}$ es la potencia de la señal de audio y P_{ruido} es la potencia del ruido mezclado a la señal.

$$SNR = 10log_{10}\left(\frac{P_{señal}}{P_{ruido}}\right) \tag{3.7}$$

3.3.4. Análisis de sensibilidad

Es importante tener un método para cuantizar nuestros resultados. Este método tiene que identificar cual de las HA bajo prueba es más robusta. Aquí se describe un método basado en un análisis de curvas ROC (acrónimo de Receiver Operating Characteristic). Este análisis permite comparar la HA de un segmento de audio degradado, con la HA completa de una pieza musical usando la distancia Hamming. Cuando se compara la HA de la versión degradada de una canción con la HA de su versión sin degradar, si la distancia entre huellas es menor que un umbral, esto se considera un Verdadero Positivo (VP). Si la comparación entrega una distancia más grande que el umbral, entonces se tiene un Falso Negativo (FN). Ahora, cuando se compara la HA de dos diferentes canciones y la distancia entre ellas es menor que el umbral, esto se considera un Falso Positivo (FP). Por último, si la distancia es mayor que el umbral, entonces se tiene un Verdadero Negativo (VN). En base a estas definiciones se pueden calcular la Tasa de Predicción Verdadera, TPV, la cual está definida por la ecuación (3.8) y la Tasa de Predicción Falsa, TPF, que se define por medio de la ecuación (3.9). Usando TPV y TPF para diferentes umbrales se puede construir la curva ROC que permite medir la sensibilidad que tiene un sistema para reconocer las señales de audio.

$$TPV = \frac{VP}{VP + FN} \tag{3.8}$$

$$TPF = \frac{FP}{FP + VN} \tag{3.9}$$

Es importante mencionar que en los experimentos el umbral toma valores entre 0 y 1200 ya que se consideran huellas binarias con esa cantidad de bits (5 segundos de audio). En las siguientes sub-secciones se proporcionan los experimentos y resultados que cuantifican la robustez respecto al ruido de ambos tipos de HA.

3.3.5. Experimento I

En este experimento se considera únicamente a HAEC1. El propósito de este experimento es encontrar el tamaño de ventana que logre el mejor o al menos un aproximado estimado de la función de densidad de probabilidad, $p(x)$, para que el desempeño de esta huella sea el deseable durante el proceso de identificación de audio. En el método de estimación de ventanas de Parzen el tamaño de ventana es un parámetro libre, por lo tanto, el estimado de la función $p\,(x)$ depende de este parámetro. Para esta huella se usa la función gaussiana como ventana del método de estimación, de esta manera, para el caso unidimensional, el valor de la varianza de la función gaussiana será el parámetro que determine el tamaño de la ventana.

Es necesario probar diferentes valores de varianza para evaluar el desempeño de HAEC1 en función de $p\,(x)$. Para este experimento se uso un conjunto de 300 piezas de música monofónica y un conjunto de 500 piezas de música polifónica. A partir de estos conjuntos, HAEC1 fue determinada para cada pieza. Después, por cada conjunto se seleccionaron aleatoriamente 65 y 50 piezas de música monofónica y polifónica, respectivamente. A continuación, extractos de audio de cinco segundos fueron tomados aleatoriamente de las señales de estas piezas, para distorsionarlos con el proceso de degradación antes mencionado y tomar sus HAEC1.

El experimento consistió en hacer la búsqueda de las huellas de los extractos degradados dentro de la colección completa de HA. Cada una de las huellas de los extractos es comparada por tramos con la huella completa de una pieza de música, haciendo un barrido de izquierda a derecha con una resolución de una trama entre cada comparación. Para determinar la posición dentro de la huella en la cual se presenta el segmento más semejante a la huella del extracto degradado se usa la distancia Hamming, es decir, el número de bits que son diferentes entre huellas. La Figura 3.7 muestra este proceso.

El resultado con respecto a la sensibilidad que tiene el sistema para reconocer extractos de piezas de música monofónica para diferentes varianzas, es presentado en la Figura 3.8. En esta figura se observan tres curvas ROC todas ellas con similar área bajo la curva y poco separadas de la diagonal principal (línea que va desde el extremo inferior izquierdo hacia la esquina superior derecha), lo que quiere decir que la tasa de

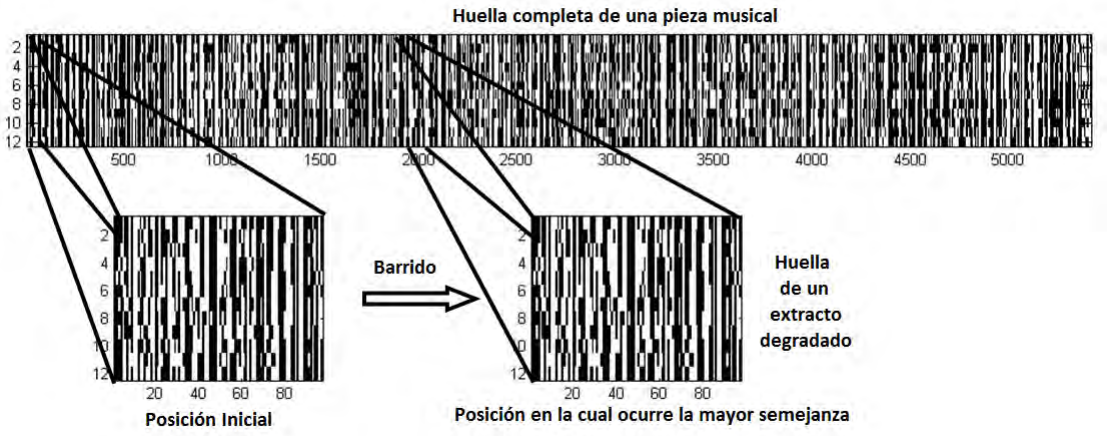

Figura 3.7: Proceso de búsqueda secuencial de la HA de un extracto degradado.

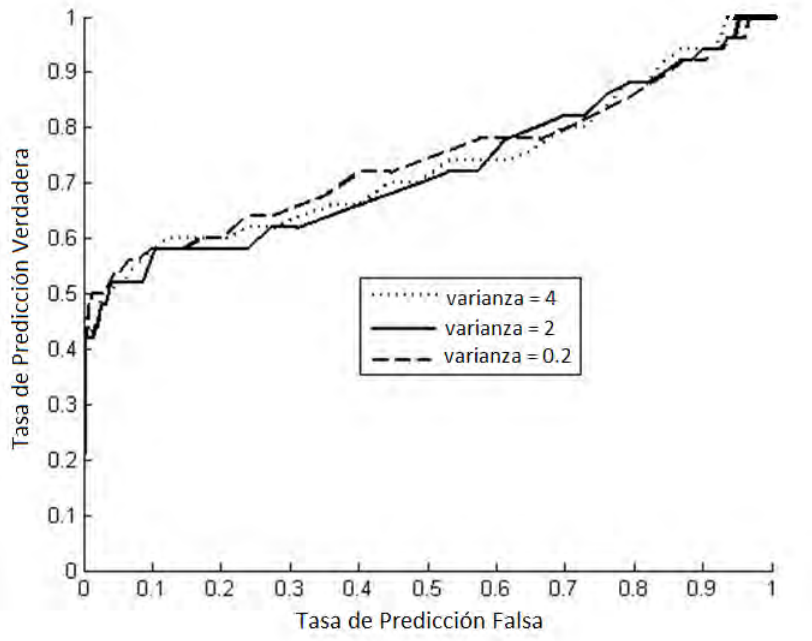

Figura 3.8: Desempeño de HAEC1 en música monofónica usando diferentes varianzas en la estimación de $p(x)$ con ventanas de Parzen.

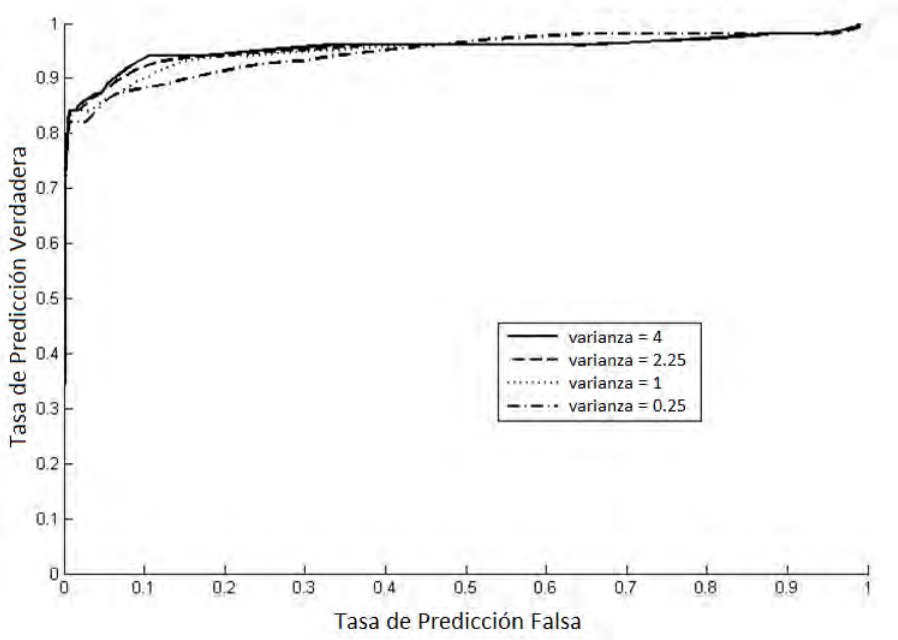

Figura 3.9: Desempeño de HAEC1 en música polifónica usando diferentes varianzas en la estimación de $p(x)$ con ventanas de Parzen.

reconocimiento del sistema no incrementa o decrementa significativamente si se varia el valor de la varianza en el rango de 0.2 a 4 y que el umbral de clasificación (el punto de la curva ROC más cercano a la coordenada [0,1]) de esta HA está muy por debajo de la coordenada [0,1] para considerla una buena alternativa para reconocer música monofónica.

En la Figura 3.9 se presenta el resultado respecto a la sensibilidad que tiene el sistema para reconocer extractos de música polifónica para diferentes varianzas. Para este tipo de música se observa que para un valor de varianza menor que 1, la curva ROC tiende a una menor área bajo la curva que si se considera una varianza mayor a 1. Sin embargo, cabe señalar que esta diferencia de áreas tampoco es muy significativa para un rango de la varianza entre 0.25 y 4, como puede apreciarse en la Figura 3.9. Por otra parte, se puede apreciar que para este tipo de música esta HA trabaja mejor al tener un umbral de clasificación más cercano a la coordenada [0,1] (o mayor área bajo la curva y más separada de la diagonal principal). En base a lo anterior, se puede concluir que el valor de la varianza (tamaño de ventana) para estimar $p(x)$, no es un parámetro que afecte de manera significativa la sensibilidad del sistema de reconocimiento por usar este tipo de HA. En el Experimento 2 se usa una varianza fija de 3 para determinar a HAEC1.

3.3.6. Experimento II

El objetivo de este experimento es evaluar la robustez a ruido que tienen los *cromagramas* convencionales y los *cromagramas de entropía*, para el caso donde las señales de audio se someten a una degradación severa. Como se explicó anteriormente, se usó el enfoque de HA para tener una representación más compacta de estas características y poder evaluar su robustez en un sistema de identificación de audio.

Para este experimento se consideran las tres huellas de audio, HAVC, HAEC1 y HAEC2. El experimento está dividido en dos partes; primero se analiza el desempeño de estas tres HA en música monofónica y después sobre música polifónica. Para la prueba con música monofónica se cree que los *cromagramas de entropía* hacen una HA más robusta que aquella que usa a los *cromagramas* como su principal característica. El experimento preliminar de la Sección 3.3.1 apoya tal afirmación y nos permite pensar que la EC ayuda a mantener los eventos de nota, aún si las señales de audio se encuentran degradadas. Para probar lo antes mencionado, considere el siguiente experimento: Se tiene un conjunto de 300 piezas de música monofónica. Por cada pieza del conjunto son extraídas las tres HA bajo prueba. A partir de las 300 piezas, 65 piezas son elegidas para extraer de ellas segmentos de 5 segundos aleatoriamente. Cada uno de estos segmentos es distorsionado con las degradaciones mencionadas en esta sección para después extraerles los tres tipos de huellas. Las HA resultantes se usan como consultas para el sistema de identificación de audio. Para determinar a qué pieza musical pertenece cada consulta se utiliza la distancia Hamming, de esta manera, se le asignará a cada consulta la pieza de audio con la que se obtenga la menor distancia en el proceso de búsqueda.

La Figura 3.10 presenta el resultado respecto a la sensibilidad que tiene el sistema para reconocer música monofónica. En esta figura se observan tres curvas ROC de las cuales la que tiene el área mayor bajo la curva y que está más alejada de la diagonal principal es la que corresponde a HAEC2. Este resultado confirma que efectivamente los *cromagramas de entropía* son más robustos que los *cromagramas* para este tipo de música. Sin embargo, la curva que corresponde a HAEC1 parece contradecir esta afirmación al presentar menor área bajo la curva que la de HAVC. Esto se debe a que el estimado de la función $p(x)$ no es lo bastante robusto si las componentes espectrales de cada *croma* son modificadas en magnitud por la adición de ruido a la señal de audio.

Para el experimento con música polifónica se usa el conjunto de 3500 piezas de audio de varios tipos de géneros. Por cada pieza del conjunto las tres huellas son extraídas y almacenadas en una base de datos. De las 3500 piezas se seleccionan aleatoriamente 350 de ellas, de las cuales se extraen segmentos de cinco segundos de audio. A continuación,

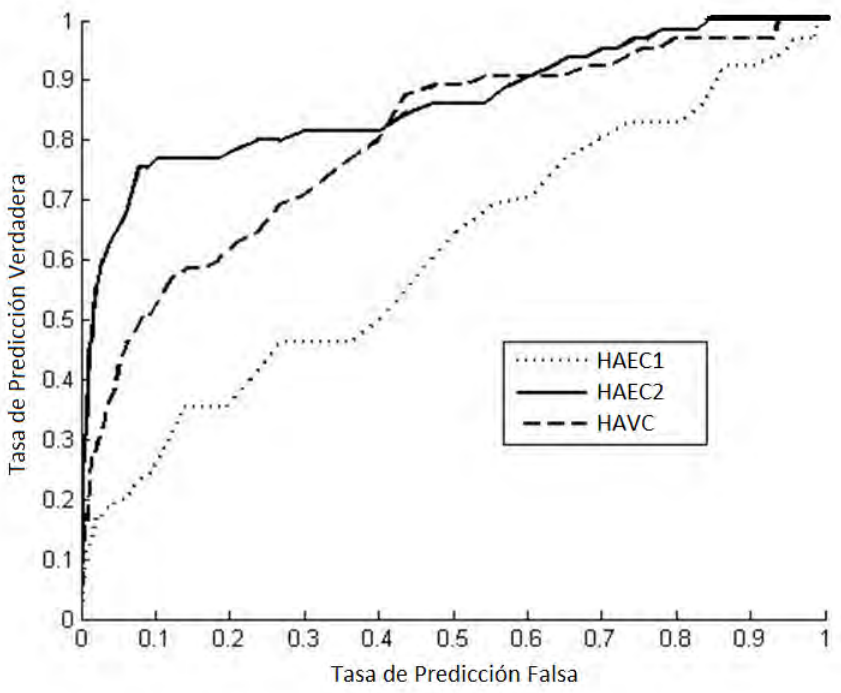

Figura 3.10: Resultados de la sensibilidad que tiene el sistema para reconocer música monofónica.

cada segmento se degrada y se le extraen sus tres tipos de huellas. Nuevamente, estas huellas serán las consultas del sistema de identificación de audio.

En la Figura 3.11 se presenta el resultado de la sensibilidad que tiene el sistema para reconocer música polifónica. Otra vez se tiene que la curva ROC con mayor área bajo la curva es la que pertenece a HAEC2. Este resultado indica que HAEC2 tuvo por mucho la mejor tasa de reconocimiento respecto al desempeño de las otras dos huellas. Una vez más se confirma que los *cromagramas de entropía* sobrepasan en robustez en cuanto a ruido a los *cromagramas* convencionales y que no importa realmente sobre qué tipo de música se utilicen, éstos siempre se desempeñarán mejor. Por otra parte, se observa que la curva de HAEC1 esta vez tuvo mayor área bajo la curva respecto a la curva de HAVC, aunque su curva sigue estando cerca de la diagonal principal. Esta prueba deja en claro que los *cromagramas de entropía* son una característica de audio que trabaja bien especialmente con música polifónica.

3.3.7. Experimento III

El objetivo de este experimento consiste en aumentar la dimensionalidad al problema de identificación de HA usando un número mayor de *valores de entropía por croma*.

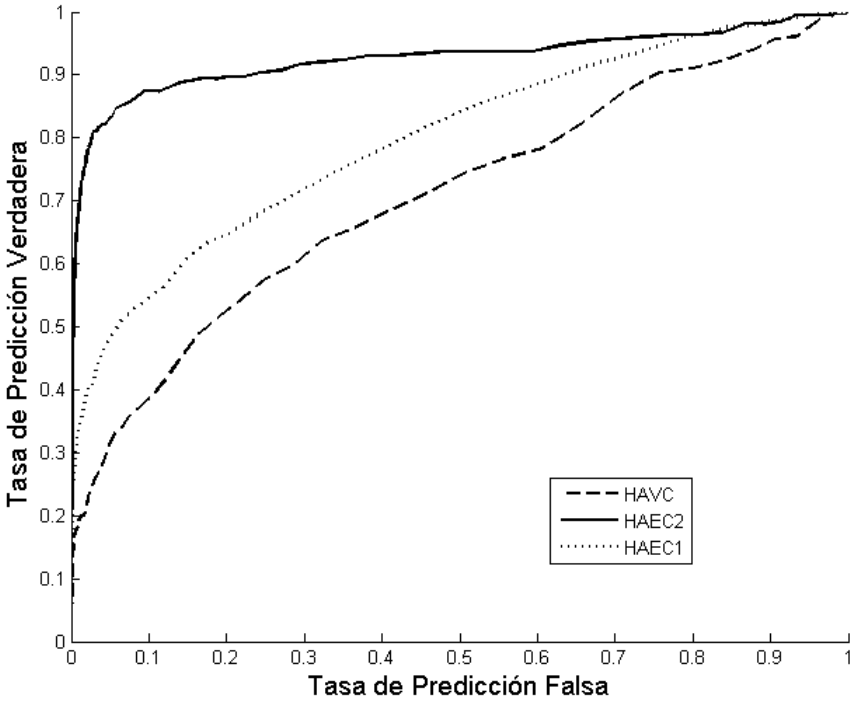

Figura 3.11: Resultados de la sensibilidad que tiene el sistema para reconocer música polifónica.

En la Sección 3.2 se estableció que el número de VEC está relacionado al número de componentes espectrales, b, que se tiene por octava a partir del espectro Q constante. Para este experimento se usan 20 valores (bits) para determinar la HA (se proponen 20 valores porque en el Capítulo 5 se requiere esa cantidad), por lo tanto, se necesitan 3 bytes para codificar cada trama de audio.

El experimento únicamente considera a HAEC2 de 12 y 20 bits, y al conjunto de 3500 piezas de música polifónica. Después de extraer y degradar 350 segmentos de audio de cinco segundos, se tomó de cada uno de ellos sus respectivas HAEC2. Al obtener los resultados del experimento, se observa en la Figura 3.12 que la curva que tiene la mayor área bajo la curva es la que corresponde a la huella de 20 bits. Este resultado indica que el sistema encontró más canciones correctamente, sin embargo, el costo computacional se incrementó al aumentar la dimensionalidad del problema. Por lo tanto, debe existir un trato entre el número de valores a usar y el tiempo de procesamiento y búsqueda del sistema, para no caer en la maldición de la dimensionalidad.

3.4. Conclusiones y Comentarios

En este capítulo mostramos que el uso de la *entropía* al lado del concepto de *valo-*

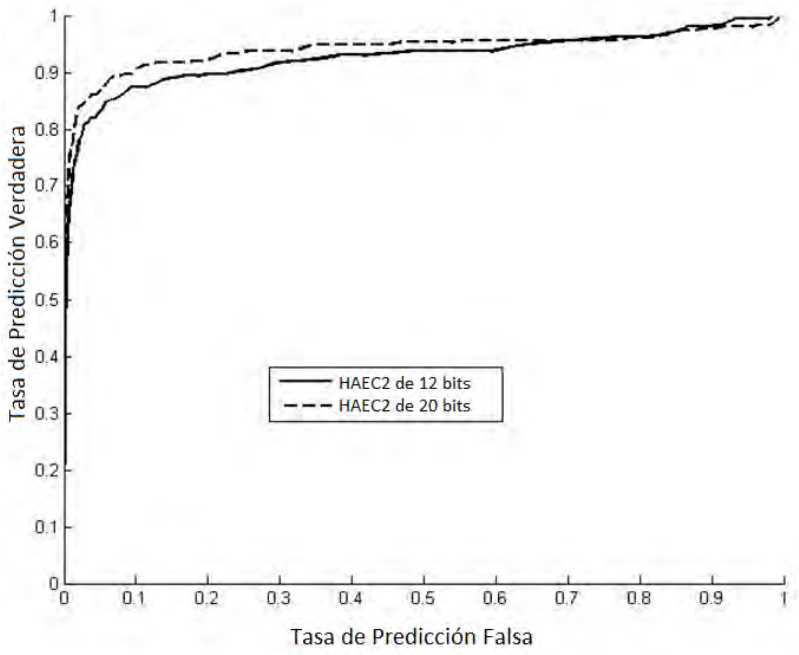

Figura 3.12: Resultado de la sensibilidad del sistema usando HAEC2 de 12 y 20 bits.

res de croma, hacen una característica que es muy robusta para señales de audio que son severamente degradadas o distorsionadas. Esta nueva característica determina la *entropía por croma* de una señal de audio y su uso se extiende no solo al reconocimiento de música monofónica, sino también al de música polifónica. La EC consiste en determinar vectores con *valores de entropía por croma* los cuales son una estimación del nivel de contenido de información por *croma* de una señal de audio. Cuando se toma una secuencia de estos vectores se forma una matriz a la cual se le puede tratar como una imagen. Los niveles de gris de esta imagen indican la distribución del *nivel de contenido de información* de cada *croma* en el tiempo. A esta imagen se le conoce como *cromagrama de entropía* e intenta mostrar al igual que los *cromagramas* cómo cambia el contenido armónico y melódico de una señal de audio. Cuando se obtiene el *cromagrama de entropía* de una pieza de audio monofónico que se degrada, los eventos de nota suelen mantenerse en este tipo de representación ubicándose en las zonas donde se tiene la más alta *entropía* (regiones oscuras).

Para determinar qué característica es más robusta a ruido, si los *cromagramas* convencionales o los *cromagramas de entropía*, se usó el enfoque de *huellas de audio* para este propósito. Tres HA fueron consideradas en nuestros experimentos; una de ellas se obtiene a partir de *cromagramas* y las otras dos a partir de la EC. Las dos HA que se determinan a partir de *cromagramas de entropía* difieren precisamente en la forma en

la que éstos son obtenidos. Uno de los enfoques usados para determinar la EC, necesita de un estimado de la función de densidad de probabilidad para determinar la *entropía*. Para este propósito utilizamos el método de estimación de *ventanas de Parzen*. El otro enfoque asume una distribución normal con media de cero, por lo tanto, se puede determinar la *entropía* usando la ecuación de la *entropía* de una variable aleatoria.

Los experimentos realizados en este capítulo involucraron música monofónica y polifónica. Usando piezas de música monofónica en el sistema de identificación de audio basado en HA, la huella que tuvo el peor rendimiento fue HAEC1. El pobre rendimiento de esta huella se debe a que el método de estimación no es lo bastante robusto cuando los datos con los que se estima la función de densidad son sometidos a ruido. Esto sin duda repercute en el valor de la *entropía* calculado. También, atribuimos este mal rendimiento a que quizá no se tengan los componentes espectrales suficientes para hacer una buena estimación de esta función. Sin embargo, la HA que tuvo el mejor rendimiento fue HAEC2, lo que indica que el método usado para determinar la EC es más robusto a la distorsión por ruido de la señal.

El experimento con piezas de música polifónica mostró que la HA con el peor rendimiento fue HAVC. Este resultado parece obvio, ya que los *cromagramas* únicamente parecen ser buenos para caracterizar música monofónica al intentar representar a las notas musicales. Por otra parte, nuevamente se tuvo que la HA con el mejor rendimiento fue HAEC2, dejando a un lado por mucho el rendimiento de las otras dos HA. Este resultado confirma una vez más que los *cromagramas de entropía* son más robustos que los *cromagramas* convencionales. Es importante notar que la HA que tiene el menor costo computacional para ser extraída es HAEC2. Finalmente, en el experimento con HAEC2 que usa 12 y 20 valores, se concluyó que debe existir un trato entre el número de valores a usar y el tiempo de extracción de éstos, ya que se puede caer en la maldición de la dimensionalidad. Sin embargo, para esta prueba se observó que usar más valores o bits hace una HA aún más robusta. Los resultados de esta aportación se encuentran publicados en las memorias de dos congresos internacionales y en la base de datos de la IEEE Xplore (ver la Sección 6.2 para más detalles).

Capítulo 4

ALINEAMIENTO POR DISTANCIA COSENO

En este capítulo se discute la aportación que tiene este trabajo en el estado del arte respecto al tema de alineamiento se series de tiempo. Este capítulo se encuentra dividido en cuatro secciones principales. La Sección 4.1 introduce al lector sobre la importancia de alinear series de tiempo. La Sección 4.2 presenta al lector la *técnica de alineamiento por distancia coseno*. La Sección 4.3 está destinada para que el lector conozca los experimentos y resultados que son obtenidos al evaluar esta técnica en un sistema de reconocimiento de voz de palabras aisladas. Por último, la Sección 4.4 expone al lector las conclusiones y comentarios de este capítulo.

4.1. Introducción

El aumento en la diversidad de tópicos que son referenciados en la literatura donde se necesita una herramienta para alinear series de tiempo, hizo que la mayoría de los investigadores se motiven para proponer nuevas estrategias para alinear series de tiempo de manera más eficiente. Como ejemplo, se puede citar el problema de alineamiento de señales electroencefalográficas donde se requiere el promediado de estas señales para realizar estudios de potenciales evocados. También se puede mencionar el problema de alineamiento de series de tiempo generadas a partir de procesos estocásticos con ruido para modelar tendencias o pronósticos. Otro ejemplo interesante es cuando una persona tiene que decir una palabra o código para acceder a un lugar restringido. Las diferentes formas de interpretar la elocución de la palabra, hará que ésta nunca sea igual a las demás elocuciones. Por lo tanto, entre elocuciones habrá diferencias temporales (duración) y morfológicas (amplitudes). Así, un sistema de identificación de voz debe

comparar la señal de una elocución, con aquellas que estén almacenadas en su base de datos. Si el proceso de reconocimiento proporciona una medidad de similitud grande al hacer el alineamiento entre la palabra que se dijo y la palabra clave de acceso, entonces, la persona podrá acceder al lugar restringido.

Por otra parte, una señal de audio se puede considerar una serie de tiempo si su secuencia de datos se puede medir en intervalos de tiempo espaciados de manera uniforme. Además, la señal de audio debe ser reproducible, es decir, debe haber un evento que la origine indefinidamente, sin que esto llegue a significar una reproducibilidad siempre exacta de la señal. De esta manera, una serie de tiempo siempre tiene réplicas. La réplica de una serie de tiempo, por lo general, nunca es igual a la serie de tiempo original debido a que ésta puede estar distorsionada por ruido y/o escalamiento [Bailey90, Gevins97, Gulmezoglu99].

Para el problema de alineamiento en general presentamos la *Técnica de Alineamiento por Distancia Coseno* (TADC) que sirve para hacer el alineamiento entre dos series de tiempo. Esta técnica tiene sus fundamentos en la distancia Coseno. La TADC es muy sencilla de evaluar y no requiere conocimiento previo de las series de tiempo, ya que el alineamiento lo hace dato por dato mediante heurísticas. Su evaluación consiste en determinar la distancia Coseno entre dos "arreglos de soporte" que tienen la funcionalidad de la estructura de datos conocida como colas. En cada iteración del algoritmo, un nuevo dato es insertado y extraído de estos arreglos, de esta manera, el alineamiento depende de la historia de la serie que está almacenada en estos arreglos en cada iteración. El tamaño de los arreglos es un parámetro libre que está directamente relacionado con la medida de similitud entre las dos series de tiempo. El resultado del proceso de alineamiento se puede obtener de dos arreglos adicionales que se usan para guardar los datos alineados y mediante el cálculo de la distancia Coseno entre estos arreglos se puede saber la medida de similitud que hay entre las dos series de tiempo. A continuación, se presentan los pasos para implementar y evaluar esta técnica.

4.2. Técnica de Alineamiento por Distancia Coseno

La técnica de alineamiento que se introduce en esta sección tiene como base la distancia Coseno. A esta técnica se le dio el nombre de *técnica de alineamiento por distancia coseno*. La distancia Coseno se define como el coseno del ángulo entre dos vectores d-dimensionales. Sean $\boldsymbol{A}$ y $\boldsymbol{B}$, dos vectores d-dimensionales, si el ángulo entre ellos es cero, ambos vectores son paralelos o llevan la misma dirección. Cuando el ángulo es cercano a cero, la mayoría de sus componentes son iguales o tienen valores muy

cercanos entre ellos. Así, la semejanza entre dos vectores está dada por la similitud de sus componentes, esto es, $s(\boldsymbol{A}, \boldsymbol{B}) = s(a_1, a_2, ..., a_d, b_1, b_2, ..., b_d)$. Por otra parte, cuando dos vectores forman un ángulo recto, ambos vectores son perpendiculares u ortogonales. Entre más disimilares sean las componentes de estos vectores, la similitud entre ellos será aproximadamente cero. De esta manera, el ángulo θ entre dos vectores también se puede usar para determinar la similitud entre éstos. Para evaluar la distancia Coseno ambos vectores deben tener la misma dimensión. Sin embargo, la similitud entre dos vectores con diferente dimensión no se puede determinar mediante una simple medida de similitud.

En programación un vector $\boldsymbol{A}$ d-dimensional es un arreglo A que se compone de d elementos. Cada elemento es referenciado por la posición que ocupa dentro del vector. Dichas posiciones son llamadas índices y siempre son correlativos. De aquí en adelante un vector $\boldsymbol{A}$ d-dimensional se considera un arreglo que contiene en sus elementos los componentes del vector, es decir, $A = [a_1, a_2, ..., a_d]$.

Un arreglo A con d elementos se puede considerar una serie de tiempo, si sus componentes a_i son los elementos x_i de la serie de tiempo $\boldsymbol{x}$. De este modo, si A y B son dos arreglos, ambos se pueden representar mediante las series de tiempo $\boldsymbol{x}$ e $\boldsymbol{y}$, respectivamente. Ahora, si $A = [a_1, a_2, ..., a_n]$ y $B = [b_1, b_2, ..., b_m]$ donde $n \neq m$, la distancia entre A y B estará determinada a través de cualquier proceso de alineamiento entre las series de tiempo $\boldsymbol{x}$ e $\boldsymbol{y}$.

Para explicar en qué consiste la TADC, sea $\boldsymbol{x}$, la serie de tiempo original de longitud n, e $\boldsymbol{y}$, la réplica de la serie de tiempo $\boldsymbol{x}$, o también, cualquier otra serie con longitud m, donde $n \neq m$. Así, el problema consiste en alinear $\boldsymbol{x}$ con $\boldsymbol{y}$ para determinar la similitud entre las dos series de tiempo.

Antes de alinear las series de tiempo con la TADC es necesario hacer un preprocesamiento de las series. La TADC está diseñada para que el ángulo entre dos vectores se encuentre siempre entre 0 y 90 grados. Esta restricción se logra haciendo todos los componentes de los vectores positivos. Se puede usar la transformación de desplazamiento dada en la ecuación (4.1) para satisfacer la restricción anterior.

$$\boldsymbol{x}^* = \boldsymbol{x} + |min\{\boldsymbol{x}\}| + 1 \tag{4.1}$$

En la ecuación (4.1) la unidad que se añade sirve para evitar divisiones por cero en la evaluación de la distancia Coseno. La TADC también requiere de dos "arreglos de soporte" de d elementos los cuales se identifican por medio de A_0 y B_0. Ambos arreglos almacenan d datos de las secuencias de las series para evaluar el ángulo de estos datos

con la distancia Coseno. Las operaciones de inserción y extracción de datos sobre estos arreglos, se llevan a cabo tal como en las estructuras de datos de cola. De este modo, el alineamiento se va realizando dato por dato en cada iteración del algoritmo. El número de elementos d de A_0 y B_0, debe ser un valor comprendido entre 2 y n, o entre 2 y m. Finalmente, la TADC requiere de dos arreglos adicionales (arreglos dinámicos), A_x y B_x, para almacenar los datos alineados producto del proceso de alineamiento. La Figura 4.1 muestra todos los elementos que intervienen en la TADC. A continuación, se describen los pasos para evaluar la TADC.

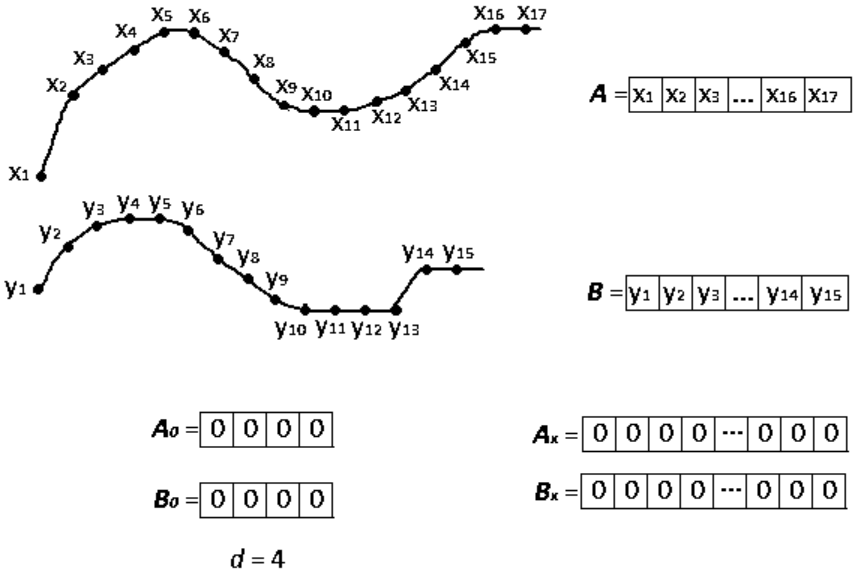

Figura 4.1: Componentes requeridos para evaluar la TADC.

1. Inicializar los arreglos A y B con los elementos de las series de tiempo $\boldsymbol{x}$ e $\boldsymbol{y}$, respectivamente. Establecer un valor para d e inicializar A_0 y B_0 con ceros. Declarar dos arreglos dinámicos, A_x y B_x, para almacenar los datos que se vayan alineando. Es necesario también definir e inicializar el ángulo de actualización, θ_i, que sirve para almacenar el menor ángulo obtenido por el proceso en cada iteración del algoritmo. Así, θ_i toma el valor inicial de 90°. Finalmente, cuatro ángulos adicionales, ϕ_1, ϕ_2, θ_1 y θ_2 serán requeridos por el proceso, los cuales tienen inicialización de $\phi_1 = 0$, $\phi_2 = 0$, $\theta_1 = 0$ y $\theta_2 = 0$.

2. Insertar directamente el elemento x_1 en los arreglos A_0 y A_x, y el elemento y_1 en los arreglos B_0 y B_x. En este paso no es necesario calcular el ángulo θ entre A_0 y

B_0 , ya que éste siempre será igual a 0 (es fácil comprobar lo anterior ya que para este caso se tiene que $\langle \boldsymbol{x}|\boldsymbol{y}\rangle = \|\boldsymbol{x}\|_2 \|\boldsymbol{y}\|_2 = 1$ y $cos^{-1}(1) = 0$). Por lo tanto, x_1 e y_1 son los primeros datos alineados del proceso y se colocan directamente sobre los arreglos A_x y B_x respectivamente.

3. A partir de este paso el ángulo θ es usado como medida de referencia para determinar qué elementos serán insertados en los arreglos A_0 y B_0. A continuación, se toman los elementos x_2 e y_2, para calcular el elemento z, que corresponde al valor medio entre x_2 e y_2. Usar la ecuación (4.2) o (4.3) para este propósito. Considere ahora los siguientes casos: Coloque z en A_0 e y_2 en B_0, y determine el ángulo θ entre A_0 y B_0, para obtener $\theta_1 = \theta$. Después, coloque x_2 en A_0 y z en B_0, y determine el ángulo θ entre A_0 y B_0, para obtener $\theta_2 = \theta$. Las condiciones mostradas en la Tabla 4.1 determinan que elementos deben ser insertados en A_0, B_0, A_x y B_x. Recuerde que el ángulo θ entre A_0 y B_0 únicamente puede variar en el rango de 0 y 90 grados al considerarse siempre componentes positivos en estos arreglos. Este paso concluye actualizando el valor de θ_i al menor de los ángulos, esto es, $\theta_i = min\{\theta_1, \theta_2\}$.

$$z = x_j - \frac{(x_j - y_k)}{2},\ si\ x_j > y_k \tag{4.2}$$

$$z = y_k - \frac{(y_k - x_j)}{2},\ si\ y_k > x_j \tag{4.3}$$

Tabla 4.1: Condiciones para determinar los elementos alineados del paso 3.

Condición	$\boldsymbol{A_0}, \boldsymbol{A_x}$	$\boldsymbol{B_0}, \boldsymbol{B_x}$
si $\theta_1 < \theta_2$ y $\theta_1 < \theta_i$	z	y_k
si $\theta_2 < \theta_1$ y $\theta_2 < \theta_i$	x_j	z

4. El proceso continúa calculando nuevamente z tal como se hizo en el paso anterior, pero ahora considerando los elementos x_3 e y_3. Una vez más los ángulos θ_1 y θ_2 deben ser calculados tomando la heurística del paso anterior. Para ambas condiciones, $\theta_1 < \theta_2$ y $\theta_2 < \theta_1$, es necesario revisar que cumpla que el menor de estos ángulos sea también menor al ángulo θ_i. Si esto es cierto, nuevamente revise la Tabla 4.1 para determinar los elementos a insertar en los cuatro arreglos. En caso contrario, una nueva consideración debe ser tomada en cuenta. Debido a que θ_i fue menor que θ_1 o que θ_2, es necesario ajustar el alineamiento considerando los elementos pasados para reducir la distancia entre ambas series de tiempo. Suponga $\theta_1 < \theta_2$ y $\theta_1 > \theta_i$. Para estas condiciones, A_0 contiene el elemento z y B_0 el elemento y_3. Considere ahora los siguientes casos: Intercambie el elemento y_3 de B_0 por el

último elemento alineado de B_x, y determine el ángulo entre A_0 y B_0, para obtener ϕ_1. Después, intercambie el elemento z de A_0 por el último elemento alineado de A_x y regrese y_3 a B_0. Determine nuevamente el ángulo entre A_0 y B_0, para obtener ϕ_2. Los ángulos ϕ_1 y ϕ_2 determinan si los elementos pasados deben repetirse para alinear de mejor manera las series de tiempo $\boldsymbol{x}$ e $\boldsymbol{y}$. Las condiciones mostradas en la Tabla 4.2 determinan que elementos se deben insertar en A_0, B_0, A_x y B_x en este paso. Los términos a_{n-1} y b_{n-1} indican los últimos elementos alineados sobre A_x y B_x respectivamente. Este paso termina actualizando el ángulo θ_i al mayor de los ángulos, esto es, $\theta_i = max\{\theta_1, \theta_2, \phi_1, \phi_2\}$.

Tabla 4.2: Condiciones para determinar los elementos alineados del paso 4.

Condición	A_0, A_x	B_0, B_x
si $\theta_1 < \theta_2$ y $\theta_1 < \theta_i$	z	y_k
si $\theta_2 < \theta_1$ y $\theta_2 < \theta_i$	x_j	z
si $\theta_1 > \theta_i$ y $\theta_1 < \phi_1$ y $\theta_1 < \phi_2$	z	y_k
si $\theta_1 > \theta_i$ y $\phi_1 < \theta_1$ y $\phi_1 < \phi_2$	z	b_{n-1}
si $\theta_1 > \theta_i$ y $\phi_2 < \theta_1$ y $\phi_2 < \phi_1$	a_{n-1}	y_k
si $\theta_2 > \theta_i$ y $\theta_2 < \phi_1$ y $\theta_2 < \phi_2$	x_j	z
si $\theta_2 > \theta_i$ y $\phi_1 < \theta_2$ y $\phi_1 < \phi_2$	x_j	b_{n-1}
si $\theta_2 > \theta_i$ y $\phi_2 < \theta_2$ y $\phi_2 < \phi_1$	a_{n-1}	z

5. Repetir el paso 4 hasta que el último elemento de las series de tiempo $\boldsymbol{x}$ e $\boldsymbol{y}$ quede alineado. En cada iteración del algoritmo se debe checar que el primer elemento que entró a A_0 y B_0, sea el primero en salir. Note que cada sumatoria de la distancia Coseno puede evaluarse más rápido si se le resta y se le suma la contribución de los elementos que se extraen y se insertan en A_0 y B_0.

6. Terminar el algoritmo evaluando la distancia Coseno sobre los arreglos A_x y B_x. Ambos arreglos representan las series de tiempo alineadas, por lo tanto, son del mismo tamaño y puede calcularse $s(A_x, B_x)$ para determinar la similitud entre las series de tiempo $\boldsymbol{x}$ e $\boldsymbol{y}$. En la Tabla 4.3 se presenta el algoritmo completo para evaluar TADC.

Es importante mencionar que la TADC cumple con las tres propiedades de una función de similitud $s(\boldsymbol{x}, \boldsymbol{y})$, esto es, $0 \leq TADC(\boldsymbol{x}, \boldsymbol{y}) \leq 1$, $TADC(\boldsymbol{x}, \boldsymbol{x}) = 1$ y $TADC(\boldsymbol{x}, \boldsymbol{y}) = TADC(\boldsymbol{y}, \boldsymbol{x})$.

4.3. Experimentos y Resultados

En esta sección se proporcionan los experimentos y resultados que son obtenidos

Tabla 4.3: Algoritmo para evaluar la TADC.

Algoritmo: Alinea dos *series de tiempo* $\boldsymbol{x}$ e $\boldsymbol{y}$ de diferentes longitudes.

Entradas: $\boldsymbol{x}$, $\boldsymbol{y}$, d. ***Salidas:*** $\boldsymbol{A_x}$, $\boldsymbol{B_x}$, $d_{\boldsymbol{A_x B_x}}$

Inicializaciones: $n = tamaño(\boldsymbol{x})$, $m = tamaño(\boldsymbol{y})$, $\boldsymbol{A_0} = \boldsymbol{0}$, $\boldsymbol{B_0} = \boldsymbol{0}$
$\theta = 0$, $\theta_i = 90$, $\theta_1 = 0$, $\theta_2 = 0$, $\phi_1 = 0$, $\phi_2 = 0$, $i = 1$, $j = 1$, $k = 1$, $l = 1$

mientras $j < n$ y $k < m$ ***hacer*** {
si $i = 1$ ***entonces*** {
$\boldsymbol{A_0}(l) \leftarrow x_j$, $\boldsymbol{B_0}(l) \leftarrow y_k$, $\boldsymbol{A_x}(i) \leftarrow x_j$, $\boldsymbol{B_x}(i) \leftarrow y_k$
$j++$, $k++$, $l++$
}
si $i > 1$ ***entonces*** {
si $x_j > y_k$ ***entonces*** $\{z = x_j - \frac{x_j - y_k}{2}\}$ ***si no*** $\{z = y_k - \frac{y_k - x_j}{2}\}$
si $l > d$ ***entonces*** $\{l = 1\}$
$\boldsymbol{A_0}(l) \leftarrow z$, $\boldsymbol{B_0}(l) \leftarrow y_k$, $\theta_1 = d(\boldsymbol{A_0}, \boldsymbol{B_0})$
$\boldsymbol{A_0}(l) \leftarrow x_j$, $\boldsymbol{B_0}(l) \leftarrow z$, $\theta_2 = d(\boldsymbol{A_0}, \boldsymbol{B_0})$
$\theta = min\{\theta_1, \theta_2\}$
si $\theta <= \theta_i$ ***entonces*** {
$\boldsymbol{A_0}(l) \leftarrow$ver Tabla 4.2, $\boldsymbol{B_0}(l) \leftarrow$ver Tabla 4.2
$\boldsymbol{A_x}(i) \leftarrow \boldsymbol{A_0}(l)$, $\boldsymbol{B_x}(i) \leftarrow \boldsymbol{B_0}(l)$
$j++$, $k++$, $l++$, $\theta_i = \theta$,
}
si $\theta > \theta_i$ ***entonces*** {
$\boldsymbol{A_0}(l) \leftarrow$ver Tabla 4.2, $\boldsymbol{B_0}(l) \leftarrow \boldsymbol{B_0}(l-1)$, $\phi_1 = d(\boldsymbol{A_0}, \boldsymbol{B_0})$
$\boldsymbol{A_0}(l) \leftarrow \boldsymbol{A_0}(l-1)$, $\boldsymbol{B_0}(l) \leftarrow$ver Tabla 4.2, $\phi_2 = d(\boldsymbol{A_0}, \boldsymbol{B_0})$
si $\theta <= \phi_1$ ***y*** $\theta <= \phi_2$ ***entonces*** {
$\boldsymbol{A_0}(l) \leftarrow$ver Tabla 4.2, $\boldsymbol{B_0}(l) \leftarrow$ver Tabla 4.2
$\boldsymbol{A_x}(i) \leftarrow \boldsymbol{A_0}(l)$, $\boldsymbol{B_x}(i) \leftarrow \boldsymbol{B_0}(l)$
$j++$, $k++$, $l++$
}
si $\phi_1 < \theta$ ***y*** $\phi_1 < \phi_2$ ***entonces*** {
$\boldsymbol{A_0}(l) \leftarrow$ver Tabla 4.2, $\boldsymbol{B_0}(l) \leftarrow \boldsymbol{B_0}(l-1)$
$\boldsymbol{A_x}(i) \leftarrow \boldsymbol{A_0}(l)$, $\boldsymbol{B_x}(i) \leftarrow \boldsymbol{B_0}(l)$
$k++$, $l++$
}
si $\phi_2 < \theta$ ***y*** $\phi_2 < \phi_1$ ***entonces*** {
$\boldsymbol{A_0}(l) \leftarrow \boldsymbol{A_0}(l-1)$, $\boldsymbol{B_0}(l) \leftarrow$ver Tabla 4.2
$\boldsymbol{A_x}(i) \leftarrow \boldsymbol{A_0}(l)$, $\boldsymbol{B_x}(i) \leftarrow \boldsymbol{B_0}(l)$
$j++$, $l++$
}
$\theta_i = max\{\theta, \phi_1, \phi_2\}$
}
}
}
i++
}
$d_{\boldsymbol{A_x B_x}} = s(\boldsymbol{A_x}, \boldsymbol{B_x})$ // El término $s(\boldsymbol{A_x}, \boldsymbol{B_x})$ se refiere a la distancia Coseno.
Nota: el término $d(\boldsymbol{A_0}, \boldsymbol{B_0})$ se refiere al ángulo de la distancia Coseno
regresar $d_{\boldsymbol{A_x B_x}}$, $\boldsymbol{A_x}$, $\boldsymbol{B_x}$

al evaluar la TADC en un sistema de identificación de voz de palabras aisladas. En principio se da una breve introducción a los sistemas de reconocimiento de voz. Posteriormente, se explica cómo está constituida la base datos donde se almacenan las palabras procesadas. Finalmente, se presentan tres experimentos. Para comprender en que consisten estos experimentos considere el siguiente problema:

Dada una pareja cualquiera de series de tiempo que están en el mismo dominio, se busca determinar si estas series poseen alguna similitud. Para determinar si las secuencias de las series de tiempo se parecen entre si, a veces es suficiente con usar una simple medida por distancia métrica tal como la distancia Euclidiana. Esta medida cuantifica la similitud que existe entre las dos series de tiempo. Sin embargo, hay procesos donde diferentes fuentes de variabilidad están presentes cuando se genera la serie de tiempo y sus réplicas. Por ejemplo, si se tiene un conjunto de réplicas de series de tiempo, el eje del tiempo puede estar diversamente desplazado, comprimido y expandido, en una manera compleja y no lineal para cada una de ellas. Adicionalmente, en algunas circunstancias, la escala de los datos medidos puede variar sistemáticamente entre la serie de tiempo y su réplica. Todos estos factores requieren una técnica de alineamiento para medir la similitud de las secuencias de las series de tiempo.

4.3.1. Sistema de reconocimiento de voz de palabras aisladas

En esta sección se presenta una descripción detallada de un sistema básico de identificación de voz de palabras aisladas. Para la mayoría de estos sistemas, la técnica de alineamiento juega un papel muy importante, ya que ésta determina si la información que se está proporcionando al sistema se ajusta o no con la información almacenada en su base de datos. Los sistemas de identificación de voz usan métodos basados en modelos del tracto vocal, análisis espectral y transformaciones lineales, entre otros, para determinar las características más relevantes de la señales de voz. Los métodos más comunes encontrados en la literatura son los siguientes:

- *Espectrogramas*. Son gráficas tridimensionales que representan la variación de la energía del contenido en frecuencia a lo largo del tiempo. Los espectrogramas se determinan a partir del espectro de una señal de tramas enventanadas. Generalmente, los espectrogramas se usan para visualizar los timbres vocálicos, la detección de formantes y las características acústicas de consonantes.
- *Codificación de predicción lineal*. Este método estima los parámetros de la voz básicos, es decir, tono, formantes, espectro y funciones del área del tracto vocal.

Además, permite representar la voz de forma compacta para almacenaje o transmisiones a baja tasa de bits [Rabiner78].

- *Coeficientes Cepstrales de Frecuencia de Mel* (MFCC por sus siglas en inglés de Mel Frequency Cepstral Coefficients). Este método está basado en la escala de frecuencias no lineal de la percepción auditiva humana. Los MFCC utilizan dos tipos de filtros, es decir, filtros espaciados linealmente y filtros espaciados logarítmicamente. Para capturar las características fonéticas más importantes de la voz, la señal se expresa en la escala de frecuencia de Mel [Logan00]. El sistema de identificación de voz presentado en este trabajo está basado en este método.

En este trabajo se procesa la forma pura de las señales de voz para representarlas como series de tiempo. Para ello, utilizamos un procedimiento de extracción de características para representar la señal de audio en un conjunto de series de tiempo. De esta manera, el problema de reconocimiento consiste en alinear el conjunto de series de tiempo de una señal con el conjunto de series de tiempo de otra señal. El procedimiento para obtener las series de tiempo de las señales de voz consiste en dos procesos, un proceso de segmentación y un proceso de extracción de características.

4.3.1.1. *Segmentación*

En la Figura 4.2 se muestra el registro de la señal de voz de la palabra "*processing*". En esta figura se puede observar como varía la amplitud de la señal de voz apreciablemente con el tiempo. En particular, la amplitud de un segmento no vocalizado es generalmente más pequeña que la amplitud de un segmento vocalizado. Para extraer la palabra "*processing*" de la señal de voz, se usa la tasa de cruces por cero, Z, y la energía, E, de la señal. Z y E identifican segmentos no vocalizados y vocalizados, respectivamente.

El análisis de Z es usualmente hecho bajo una estructura de tiempo corto. Z_n se define mediante las ecuaciones (4.4), (4.5) y (4.6),

$$Z_n = \sum_{m=-\infty}^{\infty} \left| sgn\left[x\left(m\right)\right] - sgn\left[x\left(m-1\right)\right]\right| w\left(n-m\right) \tag{4.4}$$

donde

$$sgn\left[x\left(n\right)\right] = \begin{cases} 1, & x\left(n\right) \geq 0 \\ -1, & x\left(n\right) < 0 \end{cases} \tag{4.5}$$

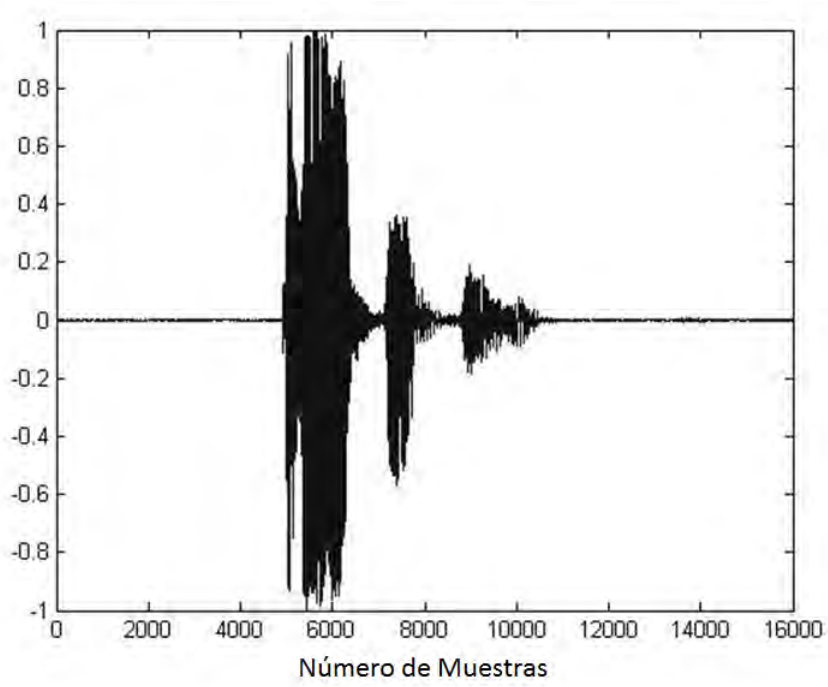

Figura 4.2: Registro de la señal de voz de la palabra "*Processing*".

y

$$w(n) = \begin{cases} \frac{1}{2N}, & 0 \leq n \leq N-1 \\ 0, & \text{por otro lado} \end{cases} \tag{4.6}$$

donde N es el número de muestras en una trama. La energía de tiempo corto, E_n, de una señal de voz proporciona una representación conveniente que refleja sonidos vocalizados. Ésta se define por medio de la ecuación (4.7),

$$E_n = \sum_{m=-\infty}^{\infty} \left[x(m)\, w(n-m)\right]^2 \tag{4.7}$$

donde $x(m)$ es la señal de voz y $w(n)$ es una ventana rectangular.

Un análisis de tiempo corto implica dividir la señal de voz en tramas. Para este sistema se usa un tamaño de trama de 30ms el cual corresponde a 240 muestras de la señal (con este tamaño de trama se asegura que no se pierdan propiedades dinámicas o variantes en el tiempo de la señal de voz). El traslape entre tramas es del 80 % (con este porcentaje de traslape se logra una correlación mayor entre tramas adyacentes). El cálculo de Z_n y E_n se hizo bajo estas consideraciones.

Se utiliza una señal de referencia, sr, para sumar las contribuciones de Z_n y E_n. La señal sr está dada por la ecuación (4.8),

$$sr = k_1 S_Z + k_2 S_E \tag{4.8}$$

donde S_Z y S_E son la tasa de cruces por cero y la energía de una secuencia de tramas, respectivamente. Las constantes k_1 y k_2 son pesos y toman los valores de 0.1 y 0.9 respectivamente. Con estos pesos se da mayor importancia a la contribución de E que a la de Z. Para extraer el segmento de audio que corresponde a la palabra, los valores de la señal sr se ajustan en amplitud dentro de un intervalo comprendido entre 0 y 1. De esta manera, si se establece un umbral de 0.2, se puede determinar el inicio y el final de cada palabra cuando la amplitud de la señal sr sea mayor que el umbral (esta condición se hace checando los valores de la señal de adelante hacia atrás y de atrás hacia adelante). La Figura 4.3 muestra un ejemplo de este proceso de segmentación.

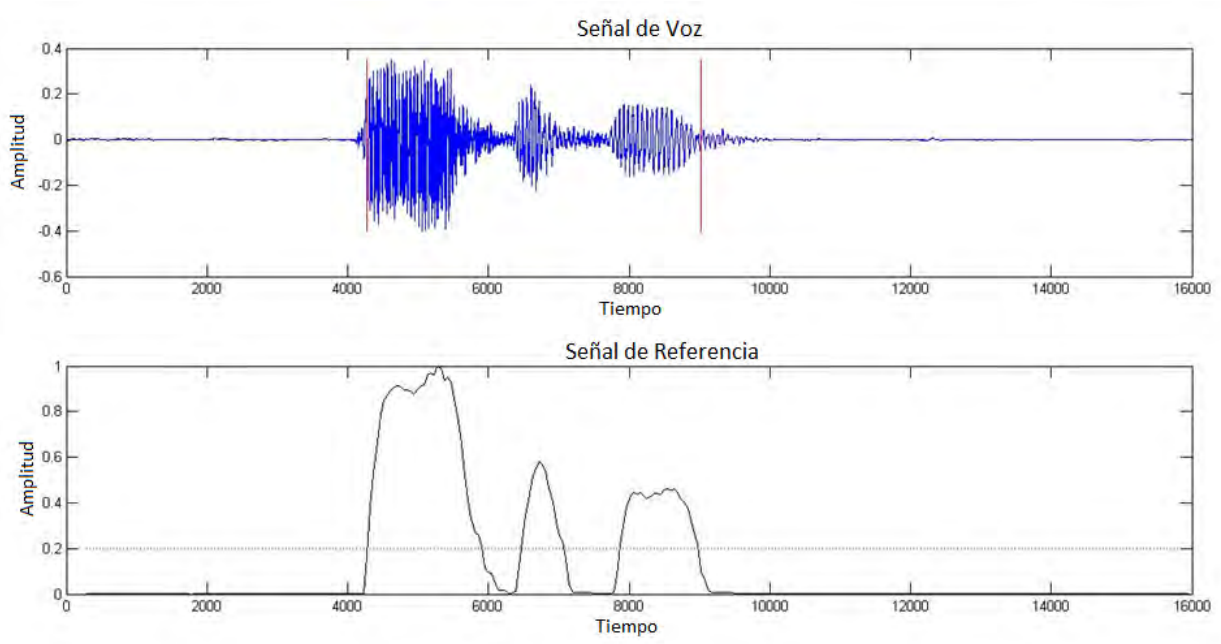

Figura 4.3: Proceso de Segmentación. Arriba se muestra el segmento de audio a extraer delimitado por medio de dos lineas verticales. Abajo se presenta la señal de referencia con el umbral de 0.2.

4.3.1.2. *Extracción de características*

El proceso de extracción de características consiste en extraer del segmento de audio los MFCC. El procedimiento comienza dividiendo el segmento de audio en tramas de 240 muestras cada una, considerando un traslape del 30 % entre tramas (con este porcentaje

de traslape se asegura que el tamaño de las series de tiempo no sean muy grandes). A cada trama se le aplica la transformada de Fourier de tiempo corto la cual está definida por la ecuación (4.9),

$$X(k) = \sum_{n=0}^{N-1} w(n)\, x(n)\, e^{\frac{-j2\pi kn}{N}} \tag{4.9}$$

para $k = 0, 1, ..., N-1$. Aquí, $x(n)$ denota los datos de audio de una trama de longitud N y $w(n)$ es la función ventana. La función ventana utilizada para este propósito fue la ventana de Hann definida en (3.2).

El espectro $|X(k)|$ es ahora escalado en frecuencia y magnitud, es decir, la frecuencia se escala logarítmicamente usando el banco de filtros, $H(k,m)$, conocido como Filtro de Mel, para después tomar el logaritmo sobre la magnitud, consiguiendo así la ecuación (4.10),

$$X'(m) = ln\left(\sum_{k=0}^{N-1} |X(k)|\, H(k,m)\right) \tag{4.10}$$

para $m = 1, 2, ..., M$, donde M es el número de filtros y $M \ll N$. El banco de Filtros de Mel es una colección de filtros triangulares definidos por las frecuencias centrales $f_c(m)$, y escritos como aparece en la ecuación (4.11),

$$H(k,m) = \begin{cases} 0, & si\ \ f(k) < f_c(m-1) \\ \frac{f(k)-f_c(m-1)}{f_c(m)-f_c(m-1)}, & si\ \ f_c(m-1) \leq f(k) < f_c(m) \\ \frac{f(k)-f_c(m+1)}{f_c(m)-f_c(m+1)} & si\ \ f_c(m) \leq f(k) < f_c(m+1) \\ 0, & si\ \ f(k) \geq f_c(m+1) \end{cases} \tag{4.11}$$

donde $f(k) = \dfrac{kf_s}{N}$ corresponde a la frecuencia del k-ésimo componente de frecuencia del espectro y f_s, es la frecuencia de muestreo. Las frecuencias centrales de cada filtro triangular son calculadas usando la escala de Mel mediante la ecuación (4.12),

$$\kappa = 2595 log_{10}\left(\frac{f}{700} + 1\right) \tag{4.12}$$

que es una aproximación comúnmente utilizada. Note que esta ecuación es no lineal para todas las frecuencias. Después, se calcula la resolución de frecuencia fija en la escala de Mel, correspondiendo al escalamiento logarítmico de la frecuencia de repetición usando la ecuación (4.13),

$$\Delta_\kappa = \frac{\kappa_{max} - \kappa_{min}}{M+1} \tag{4.13}$$

donde κ_{max} es la frecuencia de esquina superior del banco de filtros en la escala de Mel, calculada a partir de la frecuencia máxima, f_{max} usando la ecuación (4.12), y κ_{min} es la frecuencia de esquina inferior de este banco en la escala de Mel, correspondiendo a la frecuencia mínima, f_{min}. Las frecuencias centrales en la escala de Mel están dadas por la ecuación (4.14),

$$\kappa_c(m) = m\Delta_\kappa \tag{4.14}$$

para $m = 1, 2, ..., M$. Para obtener las frecuencias centrales en Hertz de cada filtro triangular se usa la ecuación (4.15).

$$f_c(m) = 700\left(10^{\frac{\kappa_c(m)}{2595}} - 1\right) \tag{4.15}$$

Finalmente, los MFCC son obtenidos por medio de la transformada discreta coseno sobre $X'(m)$ dada por la ecuación (4.16),

$$c(l) = \sum_{m=1}^{M} X'(m)\, cos\left[l\frac{\pi}{M}\left(m - \frac{1}{2}\right)\right] \tag{4.16}$$

para $l = 1, 2, ..., M$, donde $c(l)$ es el l-ésimo MFCC. Así, un vector de MFCC es obtenido por cada trama de la señal. Cada coeficiente de Mel describe una serie de tiempo si se toma la secuencia de éstos de acuerdo al número de tramas analizadas. En la Figura 4.4 se observan cinco series de tiempo refiriendo a la palabra "*processing*" obtenidas a partir de cada coeficiente de Mel.

4.3.2. Base de datos

Para registrar las señales de voz se utiliza una computadora personal. Cada palabra pronunciada es registrada por medio del micrófono de la computadora y es almacenada en formato WAV. Las características de la señal capturada son: a) sin compresión, b) frecuencia de muestreo de 8000Hz, c) cuantización a 16 bits y d) formato monoaural. Antes de extraer los MFCC, al segmento de audio de la palabra, $x(n)$, se le aplica el filtro de pre-énfasis definido por $h(n) = x(n) - ax(n-1)$ para resaltar la energía en altas frecuencias usando un factor de $a = 0.9$. Los parámetros usados para extraer los MFCC son, $f_{max} = 4000$Hz, $f_{min} = 0$Hz y $M = 5$. El valor de M indica el número de MFCC a extraer de la señal por cada trama de audio, por lo tanto, cada palabra estará

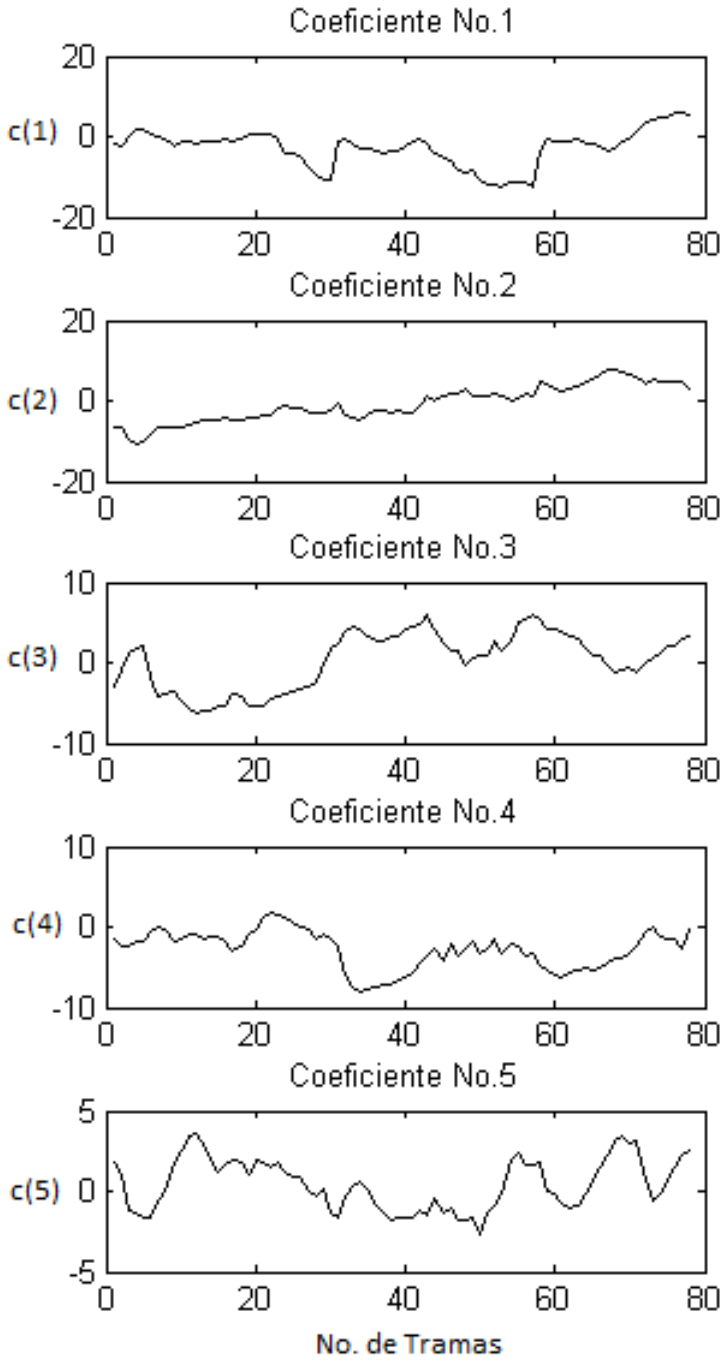

Figura 4.4: Representación de los MFCC en series de tiempo.

representada por 5 series de tiempo.

La base de datos es un archivo "TXT" en el cual se almacenan las 5 series de tiempo de cada palabra y la etiqueta de la palabra. Las 5 series de tiempo también se pueden considerar una matriz de tamaño $M \times D$, donde M es el número de MFCC y D es el número de tramas analizadas de la señal de voz. Así, cada elemento de la base de datos tiene una cabecera con la etiqueta de la palabra almacenada y en seguida la matriz con los datos de las series de tiempo.

Para los experimentos se usaron 500 palabras diferentes a las cuales se les extrajo su respectiva matriz de MFCC para almacenarlas en la base datos. La base de datos contiene palabras de diferente número de letras (duración de la elocución) y además, algunas de ellas son palabras parónimas (palabras que se parecen en su etimología) como "gato" y "pato" o "conclusión" y "confusión".

4.3.3. Experimento I

El objetivo de este experimento es evaluar el desempeño de TADC usando las series de tiempo generadas a partir de dos señales de voz. Como se explicó en la sección anterior, se usan secuencias de MFCC para generar las series de tiempo. Así, la señal de cada palabra está representada por cinco series de tiempo. Para este experimento primero se alinean las series de tiempo de dos elocuciones de la palabra "*processing*" y después, se alinean las series de tiempo de esta misma elocución con las de la elocución de la palabra "*flower*". En la Figura 4.5 se muestran sobrepuestas las cinco series de tiempo de cada elocución de la palabra "*processing*". Note como el número de tramas es diferente entre pares de series de tiempo y como hay desfasamientos entre ellas a lo largo del tiempo.

El experimento consiste en alinear las series de tiempo mostradas en la Figura 4.5 mediante TADC usando un valor arbitrario en d. El número de tramas en las series de tiempo de la palabra etiquetada por "Palabra 1" en la Figura 4.5 es de $n = 83$, mientras que para la palabra etiquetada por "Palabra 2" este número es de $m = 87$. Asignando un valor para d de 15, el resultado obtenido después de alinear las series de tiempo con TADC se observa en la Figura 4.6. Para cada coeficiente de Mel, el tamaño final para A_x y B_x resultó mayor que m y n en cada proceso, lo que quiere decir que algunos datos de la serie $\boldsymbol{x}$ y de la serie $\boldsymbol{y}$ fueron repetidos durante el proceso de alineamiento. En la Figura 4.6 se observa en la parte superior de cada gráfica el tamaño obtenido de estos arreglos después de evaluar TADC. Por otra parte, las medidas de similitud obtenidas entre cada par de series de tiempo son mostradas en la Tabla 4.4. En esta tabla

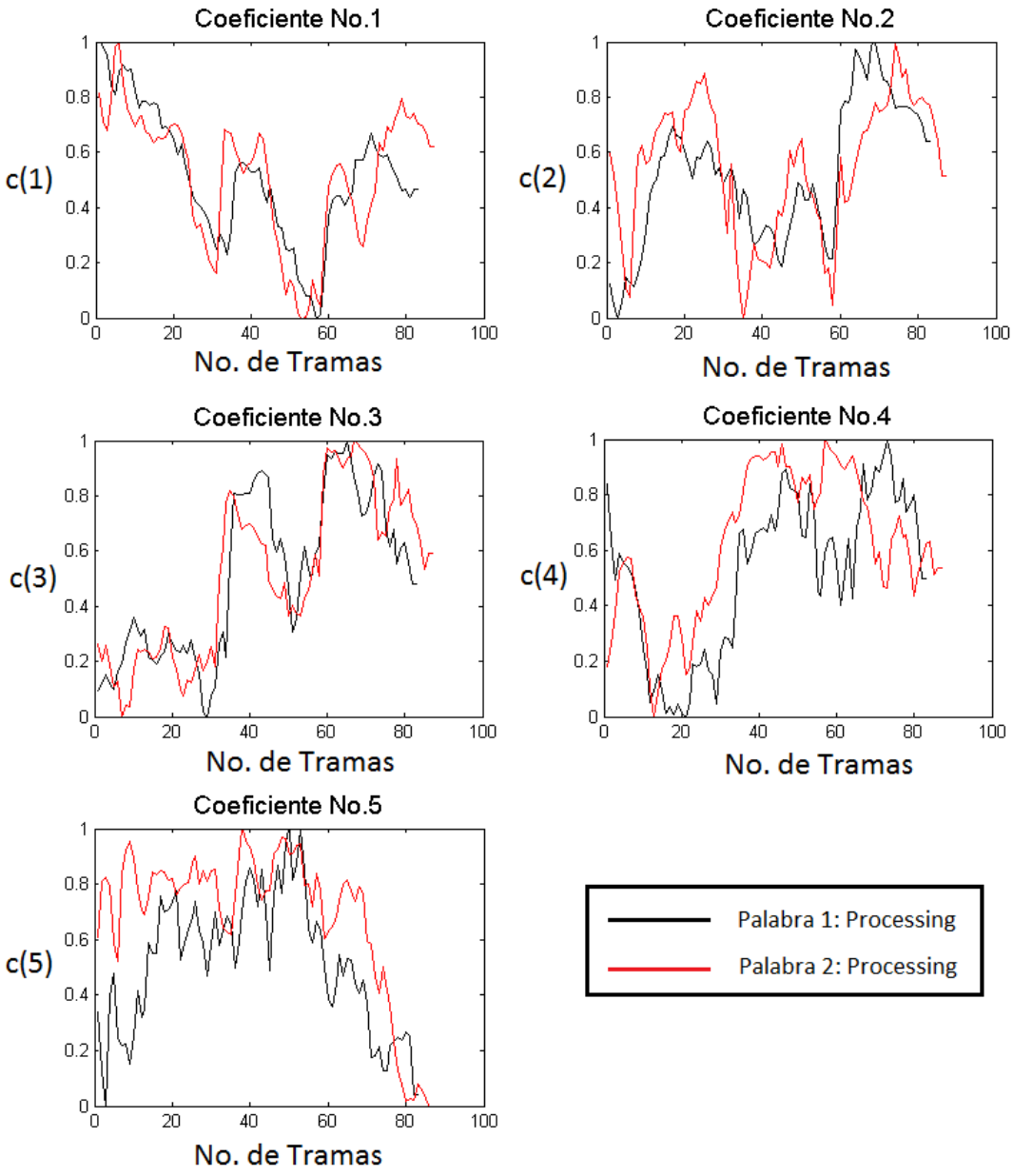

Figura 4.5: Series de tiempo sin alinear obtenidas de la palabra "*processing*".

se puede observar que todas las medidas de similitud son cercanas a la unidad. Para determinar la medida de similitud total considerando todas las medidas de similitud de la Tabla 4.4, se saca el promedio de estas medidas de similitud. Así, tenemos que para este experimento la medida de similitud total entre estas dos elocuciones es de $s(\boldsymbol{x}, \boldsymbol{y}) = \frac{0.9911+0.9851+0.9896+0.9883+0.9872}{5} = 0.9882$. Este resultado confirma que ambas

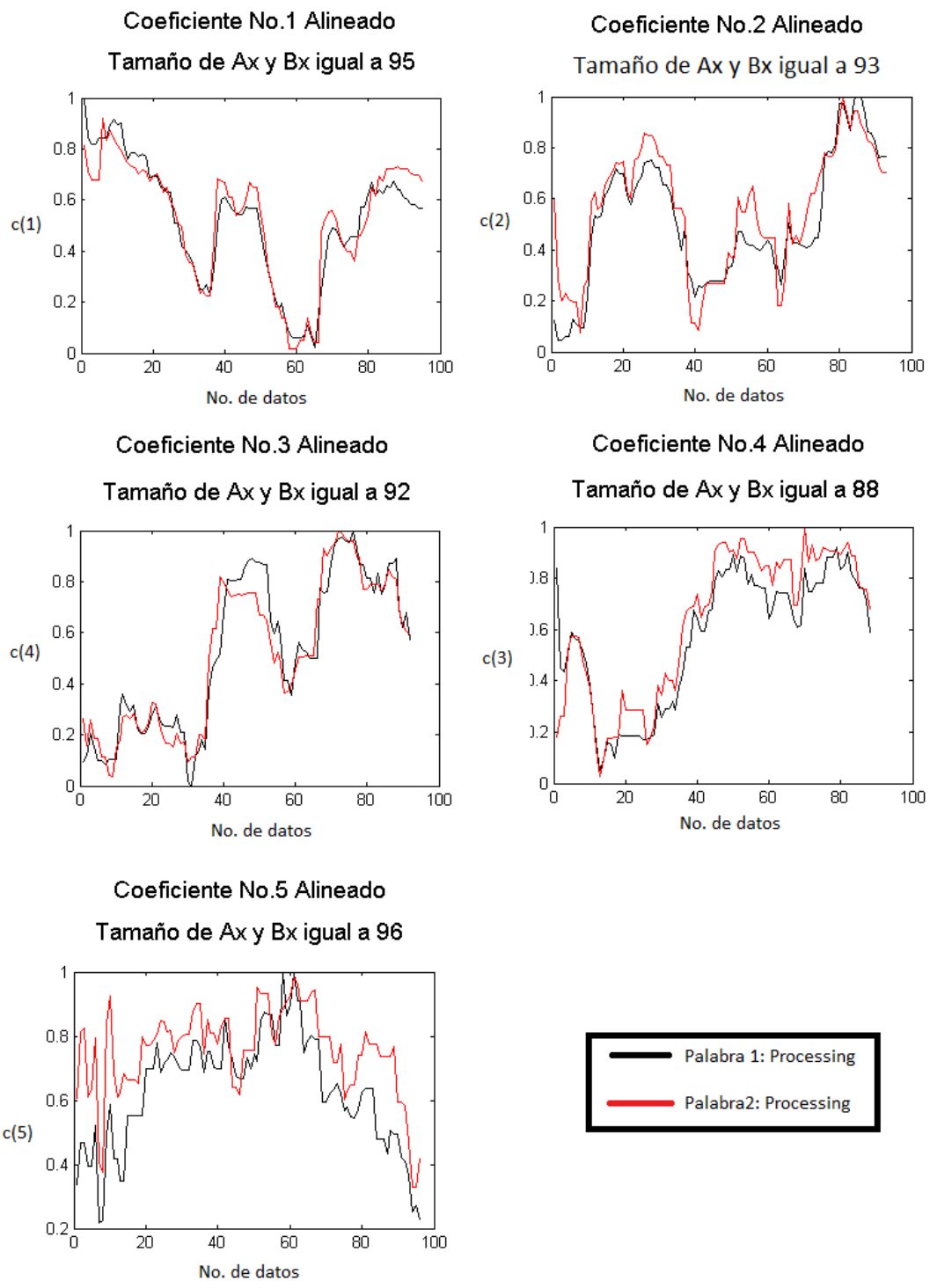

Figura 4.6: Series de tiempo alineadas por medio de TADC.

elocuciones son muy similares en su contexto.

Tabla 4.4: Medidas de similitud entre cada par de series de tiempo.

Series de Tiempo	$s(\boldsymbol{x}, \boldsymbol{y})$
Coeficiente No. 1	0.9911
Coeficiente No. 2	0.9851
Coeficiente No. 3	0.9896
Coeficiente No. 4	0.9883
Coeficiente No. 5	0.9872

Ahora, considere las series de tiempo de la elocución de la palabra "*flower*". En la Figura 4.7 se observan sobrepuestas las cinco series de tiempo de la palabra "*processing*" y de la palabra "*flower*". Se observa que los tamaños de estas series de tiempo son diferentes, en este caso se tiene que el número de tramas en las series de tiempo de la palabra etiquetada por "Palabra 1" en la Figura 4.7 es de $n = 87$, mientras que para la palabra etiquetada por "Palabra 2" este número es de $m = 66$ (n correspondiendo al número de tramas de la palabra "*processing*" y m al número de tramas de la palabra "*flower*"). Este segundo experimento consiste nuevamente en alinear cada par de series de tiempo usando TADC para $d = 15$.

El resultado de esta prueba es presentado en la Figura 4.8. El alineamiento obtenido entre cada par de series de tiempo hacen que la distancia crezca (o disminuya la similitud). Es importante notar que el tamaño final de A_x y B_x está comprendido en un valor intermedio entre n y m. Esto es debido a que TADC no terminó de evaluar todos los datos de la secuencia de la serie de tiempo más grande. Las medidas de similitud de cada par de series de tiempo son mostradas en la Tabla 4.5. En esta tabla se aprecia que hubo una pequeña disminución en cada medida de similitud debido a la diferencia que hay entre cada par de series de tiempo. Así, para esta prueba, la medida de similitud entre la elocución "*processing*" y la elocución "*flower*" es de $s(\boldsymbol{x}, \boldsymbol{y}) = \frac{0.9374+0.9476+0.9357+0.9742+0.9781}{5} = 0.9546$.

Tabla 4.5: Medidas de similitud obtenidas para las palabras "*processing*" y "*flower*".

Series de Tiempo	$s(\boldsymbol{x}, \boldsymbol{y})$
Coeficiente No. 1	0.9374
Coeficiente No. 2	0.9476
Coeficiente No. 3	0.9357
Coeficiente No. 4	0.9742
Coeficiente No. 5	0.9781

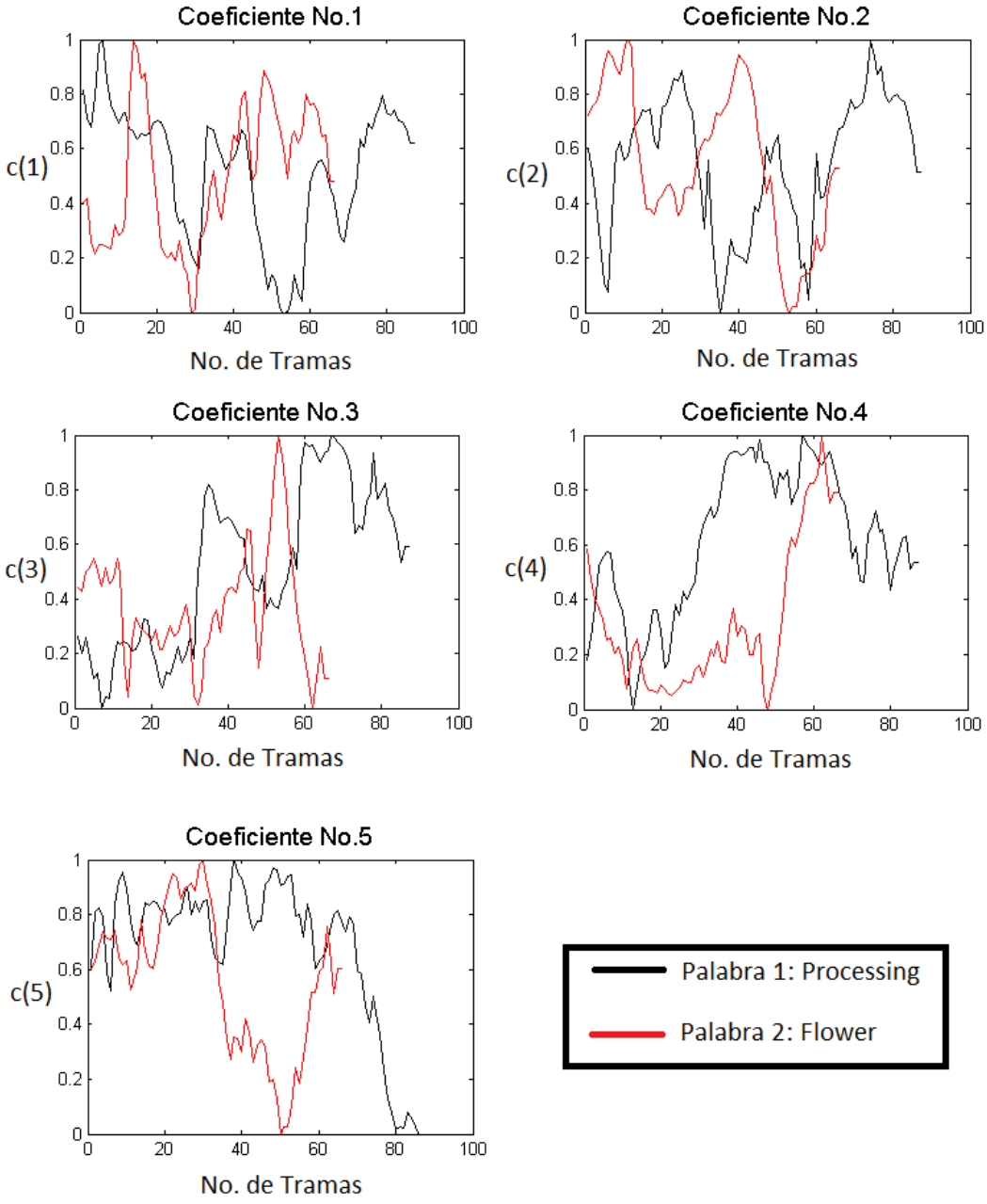

Figura 4.7: Series de tiempo sin alinear de las palabras "*processing*" y "*flower*".

4.3.4. Experimento II

Es importante evaluar el desempeño de TADC para diferentes dimensiones de los arreglos de soporte A_0 y B_0. Para este experimento se varía d en el rango de 2 hasta n, donde n es el tamaño de la serie de tiempo más corta. Nuevamente, primero se

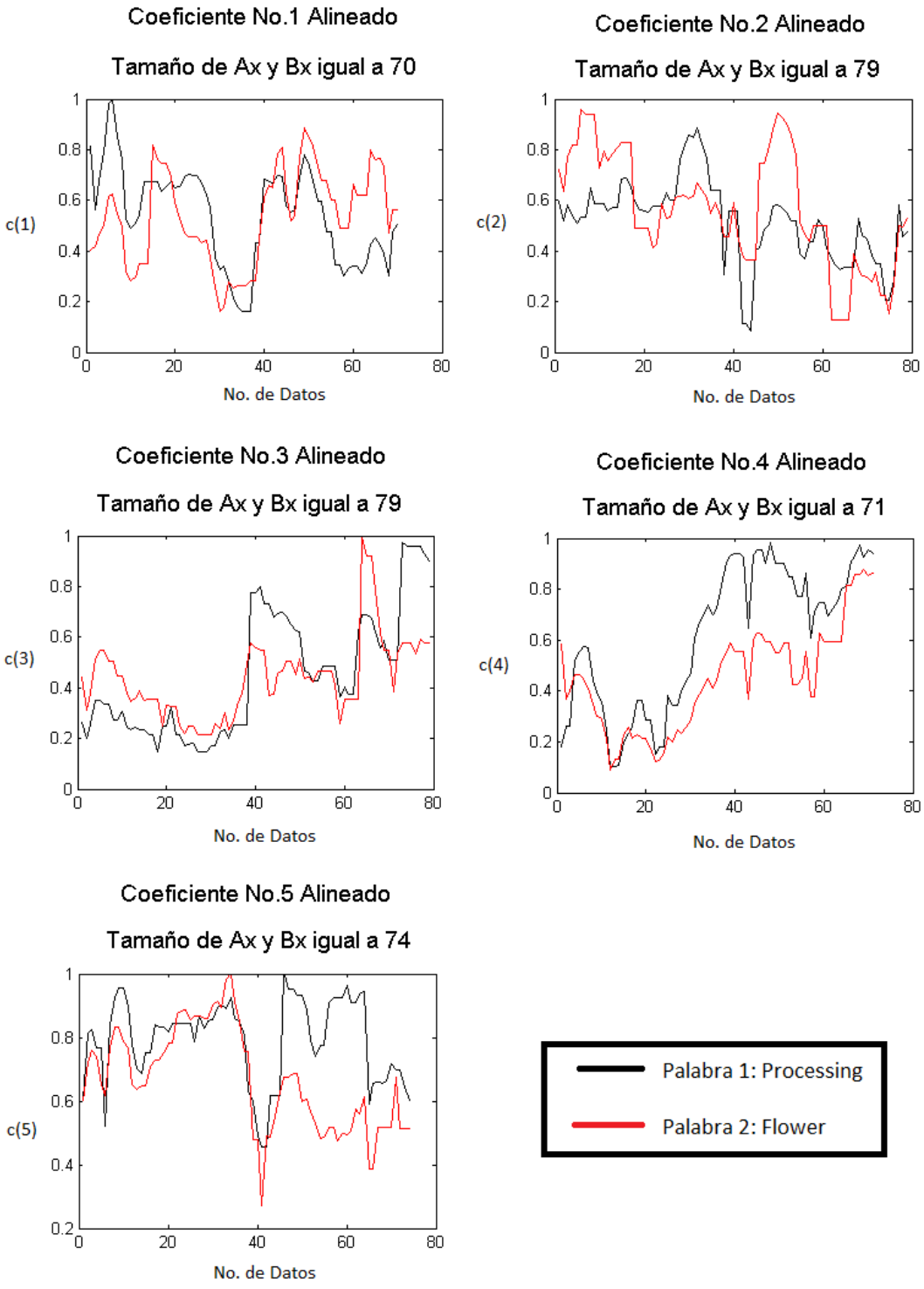

Figura 4.8: Series de tiempo alineadas de las palabras "*processing*" y "*flower*".

consideran las cinco series de tiempo de las dos elocuciones de la palabra "*processing*". Los resultados de este experimento después de evaluar TADC, son mostrados en la Figura 4.9. Estos resultados se pueden describir mejor de la siguiente manera: Para cada gráfica de la Figura 4.9, si $2 \leq d \leq \frac{n}{2}$, se presentan varaciones u oscilaciones considerables de la medida de similitud. Por otro lado, si $\frac{n}{2} < d \leq n$, estas variaciones disminuyen y la medida de similitud tiende a converger o mantenerse alrededor de un valor promedio de las medidas de similitud de este rango (únicamente si ambas series de tiempo son similares). Es importante notar que la medida de similitud es más cercana a la unidad dentro de este rango de valores para d. Ahora, cuando se consideran en este experimento las *series de tiempo* de las palabras "*processing*" y "*flower*", se llega a los siguientes resultados presentados en la Figura 4.10. De acuerdo a esta figura no se puede asumir nada acerca de la tendencia de la medida de similitud, ya que las relaciones anteriores no cumplen para todos los casos. Sin embargo, note que la medida de similitud también tiende a establecerse alrededor de un valor, aunque este valor está más alejado de la unidad.

Podemos concluir que la TADC se desempeña de mejor manera si se usan dimensiones para los arreglos de soporte en el rango de $\frac{n}{2} \leq d \leq n$. En este rango la medida de similitud es igual a un valor cercano al promedio de las medidas de similitud máximas, cuando se alinean series de tiempo similares.

4.3.5. Experimento III

El objetivo de este experimento es evaluar el desempeño de la TADC contra DTW por medio de curvas ROC [Hanley82]. El experimento consiste en reconocer el mayor número de palabras aisladas usando el sistema de identificación de voz descrito anteriormente. La base de datos de este sistema está constituida por un conjunto de 2500 series de tiempo, pertenecientes a 500 palabras diferentes. De las 500 palabras consideradas, 50 son seleccionadas para servir de consulta al sistema. De esta manera, la tarea de reconocimiento de palabras se convierte en un problema de similitud entre las secuencias de las series de tiempo.

En la implementación de DTW se utiliza la distancia Euclidiana para determinar el alineamiento entre los elementos x_i e y_i. Además, se usa la restricción de Sakoe-Chiba para encontrar la trayectoria óptima de doblado. Así, la distancia entre las series de tiempo $\boldsymbol{x}$ e $\boldsymbol{y}$, estará determinada por la distancia acumulada sobre el elemento (r, s) de la matriz de costos M. Para esta técnica también se utiliza el promedio de todas las distancias para determinar la distancia total, $d(\boldsymbol{x}, \boldsymbol{y})$, entre las series de tiempo de una

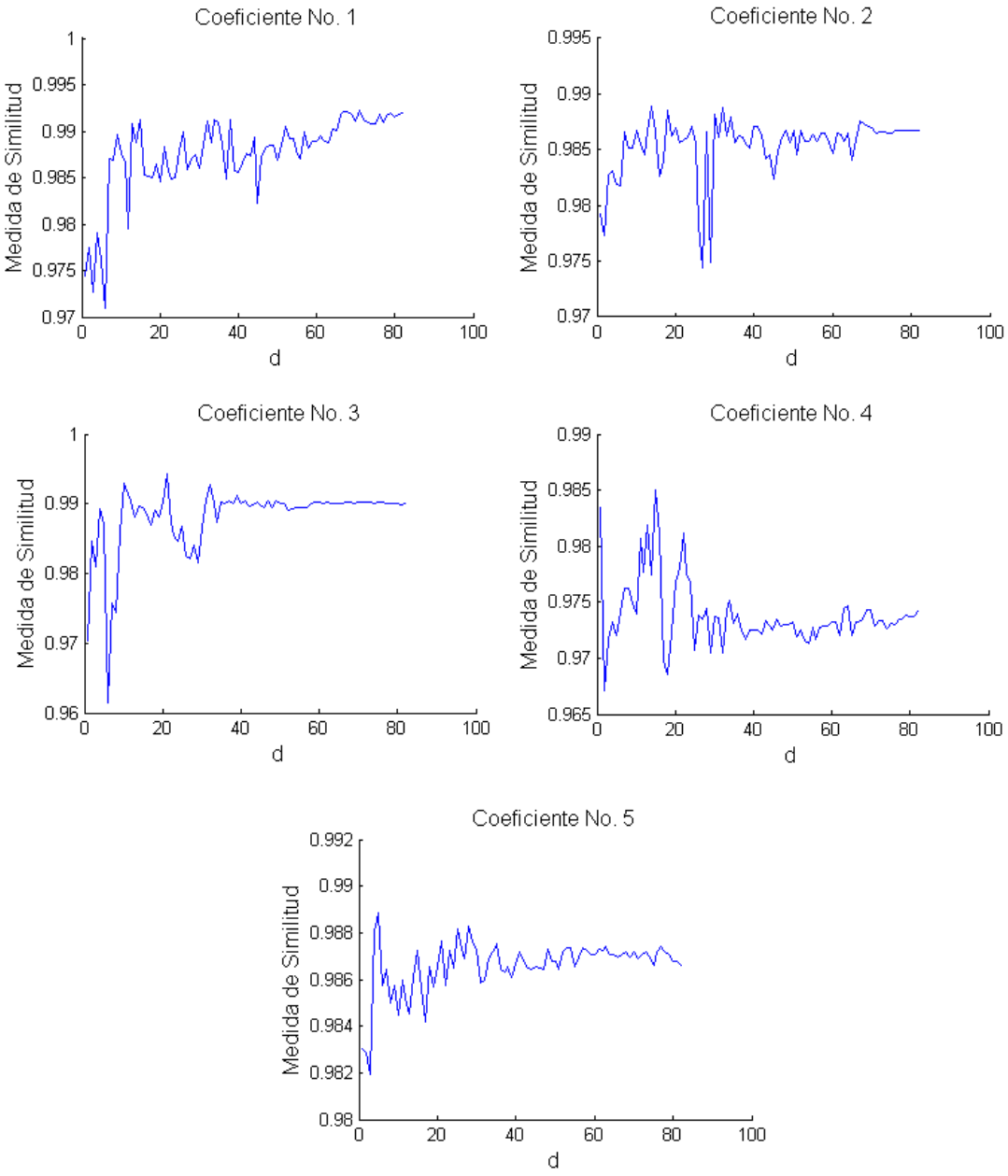

Figura 4.9: Evaluación de la TADC para varios valores de d.

elocución y otra.

Para medir el desempeño de la TADC y DTW, se evaluó la sensibilidad que tiene el sistema de identificación de voz en base a determinar la distancia entre las series de tiempo de una palabra de consulta, con las series de tiempo de las palabras almacenadas en la base de datos. Aplicando las definiciones para la Tasa de Predicción Verdadera,

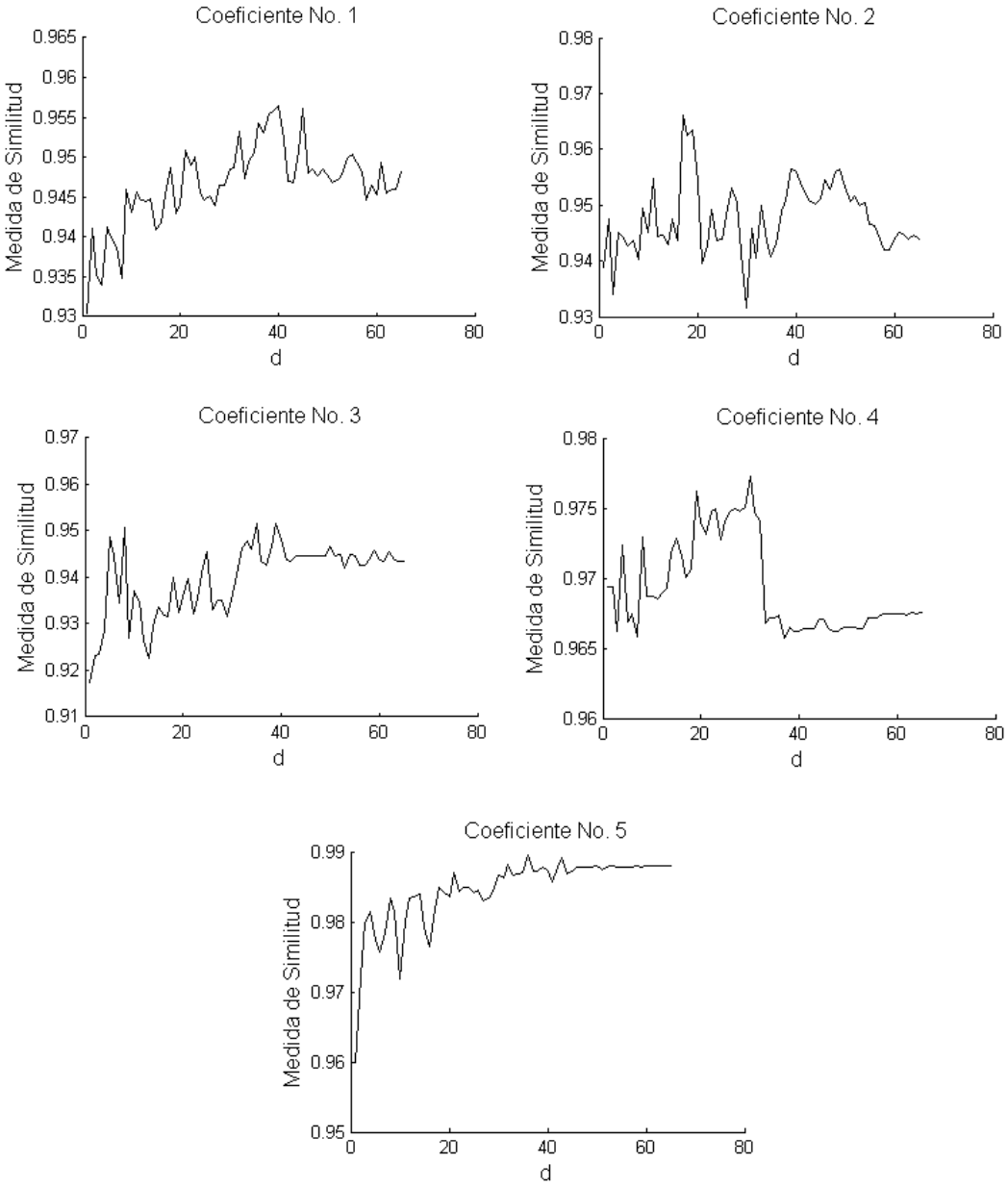

Figura 4.10: Desempeño de TADC para varios valores de d usando series de tiempo disimilares.

TPV, definida en (3.8), y la Tasa de Predicción Falsa, TPF, definida en (3.9) y por el uso de varios valores de umbral, se construyeron las curvas de sensibilidad o curvas ROC. El rango de valores dados para el umbral de este experimento fueron de 0 a 1 con incrementos de 0.01.

Para este experimento se utiliza un valor para d de $\frac{3}{4}$ del tamaño de la serie de

tiempo mayor. En la Figura 4.11 se muestra el resultado de este experimento mediante dos curvas ROC. Las dos curvas tienen aproximadamente igual área bajo la curva, lo que quiere decir que la tasa de reconocimiento de ambas técnicas es aproximadamente la misma. También, se observa que el mejor punto de sensibilidad para ambas técnicas está situado relativamente cerca de la coordenada [0, 1] del espacio ROC, de esta manera, ambas técnicas tienen un desempeño por arriba del 90 %. Con estos resultados se puede asumir que DTW se desempeña de mejor manera al tener ligeramente mayor área bajo la curva, sin embargo, la TADC tiene una ventaja sobre DTW como se demuestra a continuación.

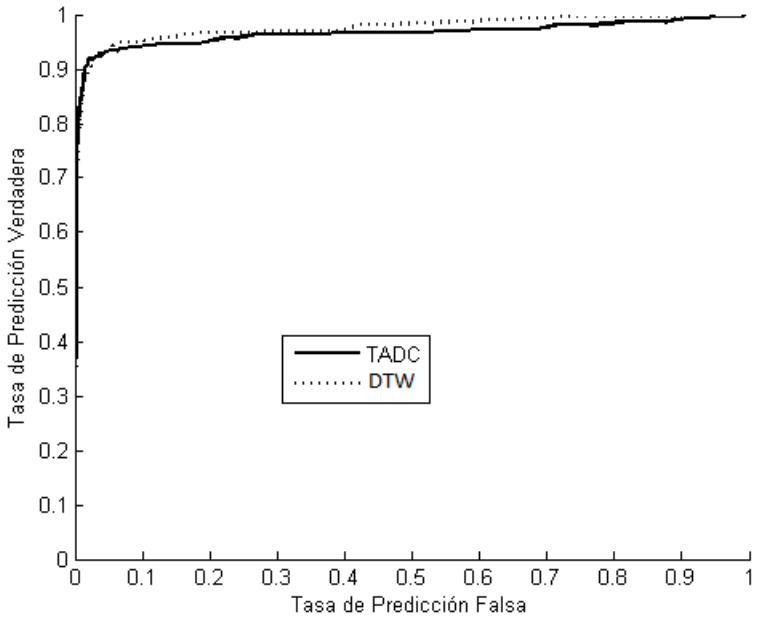

Figura 4.11: Resultados de la evaluación de la TADC y DTW.

La última prueba de este experimento consiste en evaluar el tiempo que toma cada técnica en alinear las series de tiempo para diferentes números de datos en sus secuencias. En específico, el número de datos en cada serie de tiempo se varía en el rango de 100 a 1000 datos (esto se logra si se varía el traslape entre tramas de la señal o se usa una interpolación). El resultado de esta prueba se muestra en la Figura 4.12. La gráfica de DTW muestra el conocido comportamiento cuadrático del tiempo de complejidad de este algoritmo al considerarse $r = s$. Por otra parte, la gráfica de la TADC muestra un comportamiento lineal con respecto del tiempo, por lo tanto, el tiempo de complejidad de la TADC es $O(n)$. Esto es fácil de comprobar si se analiza el algoritmo dado en la Tabla 4.3 donde se observa que no hay recursiones por asignación, por lo tanto, se

puede concluir que su costo computacional es lineal.

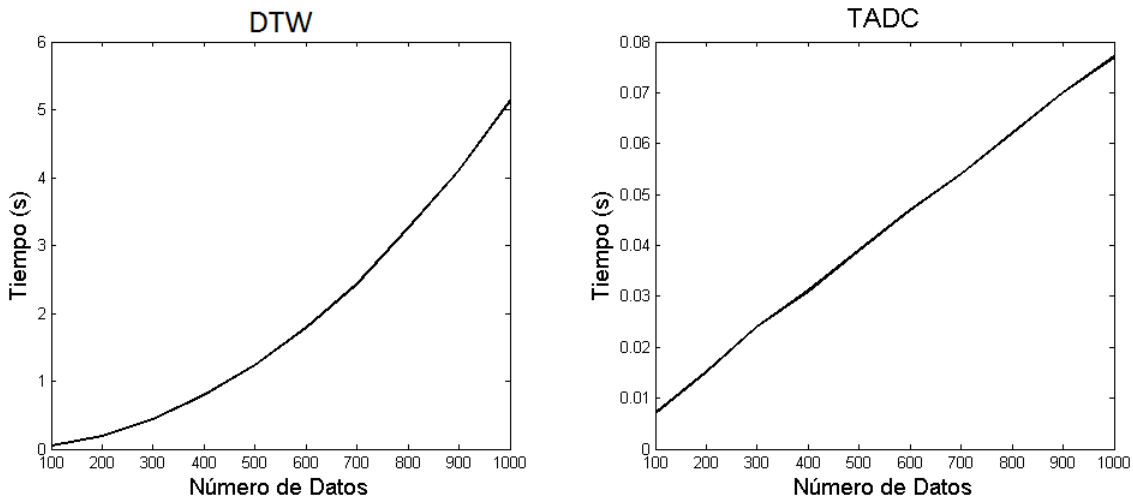

Figura 4.12: Análisis del tiempo de complejidad de los algoritmos de DTW y TADC.

4.3.5.1. *Limitaciones y Fallas*

Por otra parte, es necesario mencionar las limitaciones que tiene la TADC y los casos cuando el algoritmo puede fallar.

- **Limitaciones**: La única limitante que tiene la TADC es que no puede alinear series de tiempo muy caóticas. Un ejemplo muy claro sería si se quisiera usar esta técnica para alinear señales de audio en su forma pura. En este ejemplo, el alineamiento entre señales sería erróneo ya que la correspondencia entre las muestras (el contenido perceptual) de las dos señales no sería la correcta.
- **Fallas**: El algoritmo puede fallar por las siguientes tres circunstancias: a) Pérdida de datos en una de las series de tiempo (lo que en inglés se le conoce como "Cropping"), b) Desplazamiento excesivo en amplitud entre las series de tiempo que se están alineando, y c) Series de tiempo de diferentes tamaños, donde la parte del inicio de la serie de tiempo mayor se parece en su totalidad a la serie de tiempo menor.

4.4. Conclusiones y Comentarios

En este capítulo se presentó una nueva técnica de alineamiento de series de tiempo. Esta técnica se le denominó *Técnica de Alineamiento por Distancia Coseno*. Como su nombre lo indica, esta técnica utiliza la distancia Coseno para realizar el proceso de

alineamiento. La TADC es una técnica muy sencilla de implementar, ya que únicamente requiere de heurísticas simples. Además, el alineamiento lo realiza dato por dato utilizando dos arreglos de soporte que tienen la funcionalidad de estructuras de datos de cola. De esta manera, la TADC se puede considerar una técnica no Markoviana, al considerar la historia de las series de tiempo que es almacenada en estas colas. En cada iteración del algoritmo es insertado y extraído un dato de estas colas. Esta operación facilita el cómputo de todas las sumatorias presentes en la ecuación de la distancia Coseno. Es importante mencionar que el tamaño de las colas es un parámetro libre del cual depende la medida de similitud obtenida entre dos series de tiempo.

Para los experimentos se utilizó un sistema de identificación de voz de palabras aisladas. Este sistema utiliza los MFCC para extraer las características más relevantes de las señales de voz. El comportamiento de cada coeficiente en el tiempo describe una serie de tiempo, por lo tanto, se tienen tantas series de tiempo como coeficientes haya. Para los experimentos usamos 5 coeficientes de Mel, así, cada palabra está representada por 5 series de tiempo. En los experimentos se usa DTW para validar el desempeño de la TADC, ya que ésta es la técnica de alineamiento más referenciada en la literatura sobre este tópico.

El primer experimento consistió en evaluar el desempeño de TADC, alineando las series de tiempo de una misma palabra y las series de tiempo de dos palabras diferentes. Los resultados de este experimento usando un valor para d de 15, mostraron lo siguiente: Para la misma palabra, las medidas de similitud entre sus series de tiempo están en un valor cercano a la unidad. El promedio de estas medidas determina la medida de similitud total del proceso de alineamiento. Para esta prueba la medida de similitud total fue de 0.9882. Por otro lado, utilizando palabras diferentes la medida de similitud total fue de 0.9546. En este experimento también se comprobó que TADC cumpliera con las tres propiedades de una función de similitud.

El segundo experimento consistió en evaluar el desempeño de TADC para diferentes valores de d. Nuevamente, en este experimento se usaron las series de tiempo de una misma palabra y las series de tiempo de dos palabras diferentes. Para la misma palabra, el comportamiento de la medida de similitud fue el siguiente: Si $2 \leq d \leq \frac{n}{2}$, este rango se considera el transitorio de la medida de similitud ya que presenta oscilaciones considerables sin una tendencia a converger. Por otro lado, si $\frac{n}{2} < d \leq n$, la medida de similitud tiende a converger o mantenerse alrededor del valor promedio de todas las medidas de similitud de ese rango. En este rango la medida de similitud es lo suficientemente grande para considerarla la mejor medida de similitud entre dos series de tiempo. Para el caso donde se tienen dos palabras diferentes estas relaciones no

cumplen, por lo tanto, no se puede asumir nada acerca del comportamiento de la medida de similitud. Sin embargo, la medida de similitud es menor y esto permite discriminar entre series de tiempo disimilares.

El tercer experimento consistió en evaluar el desempeño de la TADC y DTW sobre un sistema de identificación de voz de palabras aisladas. Para esto, se utilizó la sensibilidad del sistema como parámetro de medición, el cual mostró que ambas técnicas tienen semejante tasa de reconocimiento al presentar similar área bajo la curva ROC. También es importante mencionar que ambas técnicas presentaron un desempeño por arriba del 90 %. Sin embargo, este resultado deja dudas sobre que técnica de alineamiento es mejor. Debido a esto, se realizó una última prueba, la cual consistió en determinar que técnica se evalúa más rápido para diferentes tamaños en la secuencia de las series de tiempo. El resultado de esta prueba mostró que la TADC se evalúa más rápido que DTW y que su orden de complejidad es lineal. Por lo tanto, TADC se convierte en una herramienta útil si se quiere alinear series de tiempo con secuencias de datos de tamaños grandes. Los resultados de esta aportación se encuentran publicados en las memorias de un congreso internacional y en la revista International Journal of Combinatorial Optimization Problems and Informatics (ver la Sección 6.2 para más detalles).

Capítulo 5

ALINEAMIENTO POR ESTADO DE CREENCIA

En este capítulo se discute la aportación que tiene este trabajo en el estado del arte respecto al tema de alineamiento de audio en tiempo real. Este capítulo se encuentra dividido en cinco secciones principales. La Sección 5.1 ayuda al lector a comprender la importancia que tiene el uso de la *entropía por croma* en el tópico de alineamiento de audio en tiempo real. Además, proporciona una breve introducción al tema de sistemas de seguimiento de audio. La Sección 5.2 sirve para que el lector comprenda cómo es que la *entropía por croma* también puede considerarse una característica robusta a las variaciones dinámicas de las señales de audio si se hace su descomposición en valores propios. La Sección 5.3 introduce brevemente al lector en la teoría sobre procesos de decisión de Markov y además le proporciona los fundamentos para llegar al *proceso de Markov parcialmente observable* que proponemos en este trabajo. La Sección 5.4 está destinada para que el lector conozca los experimentos y resultados preliminares sobre el alineamiento de señales de audio en tiempo real. Por último, la Sección 5.5 expone al lector las conclusiones y comentarios de este capítulo.

5.1. Introducción

En el Capítulo 3 se explicó la razón principal por la que fue diseñada la EC (*entropía por croma*). Esta razón está directamente ligada con el problema de alineamiento de señales de audio en tiempo real. Esto es importante que quede claro, ya que este capítulo está dedicado específicamente a este problema. Para este problema en especial, no todas las características de audio que se encuentran en la literatura pueden considerarse. Las propiedades que deben de satisfacer las características de audio para este problema son:

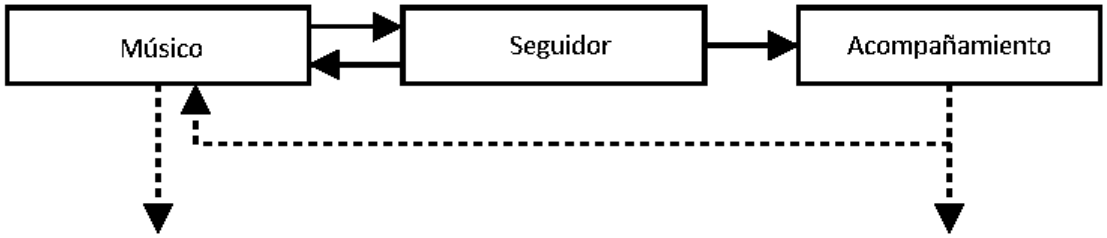

Figura 5.1: Elementos de un sistema de seguimiento de audio [Orio03].

a) inmunidad al ruido y b) robustez a las variaciones dinámicas tanto en amplitud como en tiempo de una señal. El Capítulo 3 se dedicó específicamente a probar que la EC cumple con la primera propiedad. Para probar si la EC es una característica de audio robusta a las variaciones dinámicas tanto en amplitud como en tiempo de las señales de audio, se utiliza como base un sistema de seguimiento de audio para probarlo. A continuación, se describe brevemente en que consiste un sistema básico de este tipo.

Un Sistema de Seguimiento de Audio (SSA) tiene al menos los elementos mostrados en la Figura 5.1 [Orio03]; el músico (humano), el seguidor (computadora) y el acompañamiento (también llamado interpretador automático o parte electrónica). Estos elementos interactúan unos con otros. El papel del flujo de comunicación a partir del músico a la computadora es claro, debido a que la conducta de la computadora está casi completamente basada en el comportamiento humano.

La Figura 5.2 presenta la estructura de un SSA general [Orio03]. En la figura se observa un bloque de pre-procesamiento de la señal, donde el sistema extrae algunas características de audio (ej. frecuencia fundamental, energía, espectro, amplitud, flujo cepstral, entre otras) a partir de los sonidos de audio de una interpretación. Cada SSA define un conjunto de características relevantes de la señal de audio, las cuales son usadas como descriptores de la interpretación en vivo. Estas características definen la dimensión del espacio de entrada del modelo creado a partir de la "partitura objetivo". La partitura objetivo se refiere a una partitura escrita (lista de notas) o también a la señal de audio de la partitura que el sistema tiene que seguir. Idealmente esta partitura es idéntica a la partitura que se está ejecutando. La pregunta sobre qué clase de formato de partitura es usada en la partitura objetivo, es muy importante para la ergonomía del sistema y para su desempeño.

El bloque "modelo" representa la estructura interna del sistema que interactúa con la partitura objetivo. En este bloque se encuentra el modelo el cual es equiparado con los datos de audio que entran en el sistema para determinar las "acciones de partitura",

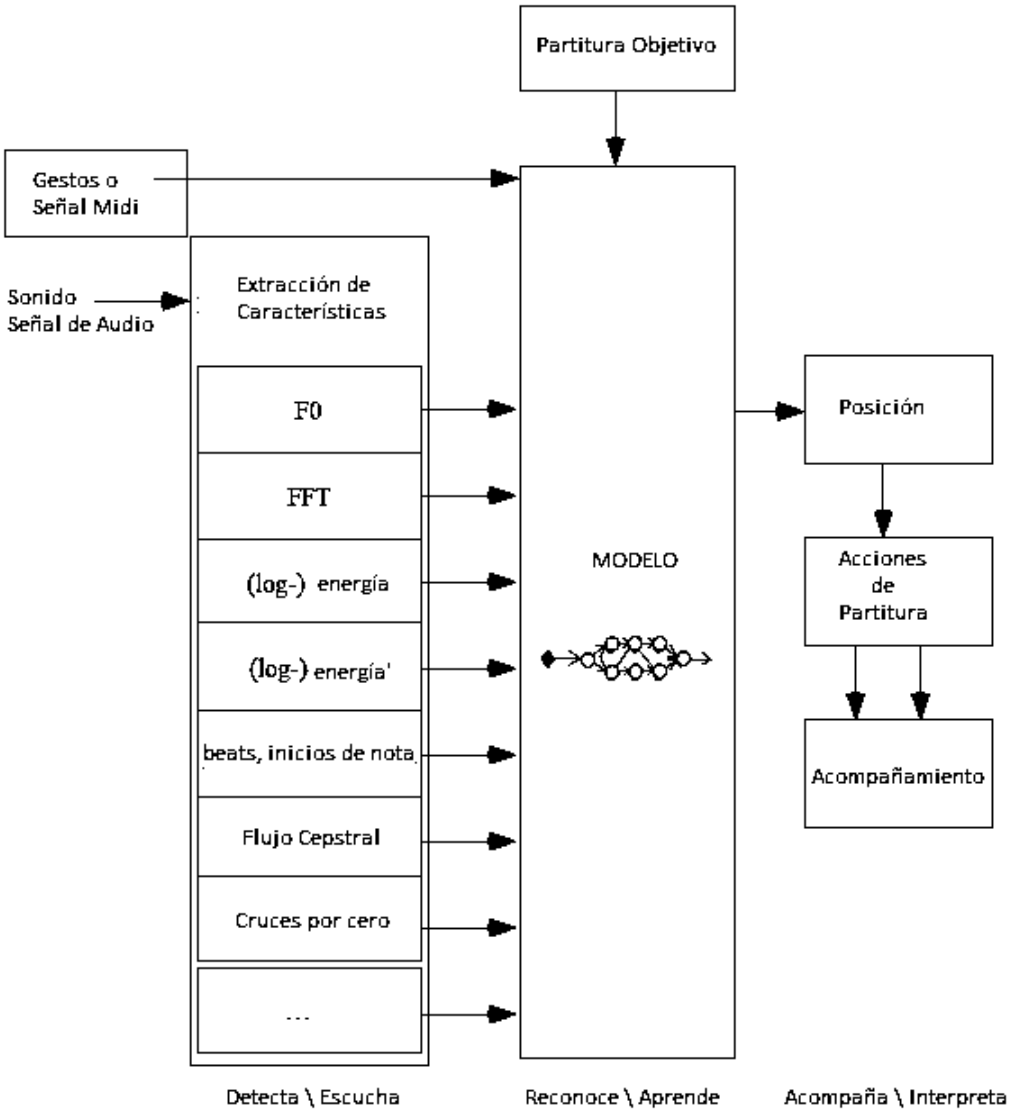

Figura 5.2: Estructura de un sistema de seguimiento de audio [Orio03].

que representan las acciones que el seguidor tiene que desempeñar en algunas posiciones específicas (ej. síntesis de sonidos, encendido de luces, juegos artificiales, etc.). El bloque "posición" determina el tiempo actual del sistema relativo a la partitura objetivo.

En todo SSA el proceso de extracción de características juega un papel muy importante, ya que de éste depende la eficiencia del sistema para reconocer la posición en el tiempo del flujo de audio que ingresa al sistema. Una mala elección del método de extracción de características llevará al sistema a tener un mal desempeño, aunque el diseño del modelo sea el adecuado. Dannenberg [Dannenberg03], por ejemplo, utiliza como características los *valores de croma* del audio generado a partir de un archivo MIDI y de la señal de audio de esa pieza tocada por músicos. Dixon [Dixon05], por otra parte, utiliza la energía de la señal en cada banda de frecuencia. Dixon hace el seguimiento alineando las señales de audio de dos interpretaciones de una misma pieza musical. Otros trabajos encontrados en la literatura usan los MFCC y la *entropía espectral multibanda* para este propósito [Camarena10].

En este trabajo se considera a la señal de audio de la interpretación de la pieza musical a seguir como la partitura objetivo. Sin embargo, no se maneja esta señal en su forma pura, sino por medio de vectores propios extraídos a partir de *cromagramas de entropía*. De este modo, la partitura objetivo es una sucesión de valores propios que representan a la señal de audio y esta representación sirve para que el sistema pueda reconocer diferentes versiones de la interpretación, a pesar de que se ejecuten con diferente instrumentación y/o con diferentes intérpretes. A continuación, se presenta el análisis que justifica las razones del porque los vectores propios extraídos a partir de *cromagramas de entropía* son usados como características robustas en la identificación de audio.

5.2. Análisis de Descomposición de Valores Propios

El proceso de extracción de características de la señal de la partitura objetivo se basa en el concepto de EC. La EC estima el nivel de contenido de información en cada *croma* de una señal de audio. Para determinar la EC se extrae por cada trama de la señal, un vector con *d valores de entropía por croma*. Una sucesión de estos vectores forma una matriz a la cual se le denomina *cromagrama de entropía*. Los *cromagramas de entropía* son robustos a distorsiones en la señal de audio y a la dinámica de la música. Para saber si esta característica es también robusta a las variaciones dinámicas de tiempo y amplitud de la música considere el siguiente análisis.

El Análisis de Descomposición de Valores Propios (ADVP) consiste en determinar

los valores propios de dos matrices para probar si cumplen con la siguiente afirmación:

De la observación, se sabe que los valores propios $\lambda \in K\,(\Re\, o\, \mathbb{C})$ de una matriz cuadrada $\boldsymbol{A}$ de orden n, son aproximadamente iguales a los valores propios de una matriz cuadrada $\boldsymbol{B}$ de orden n, si sus elementos, a_{ij} y b_{ij}, son similares. Ejemplo, si

$$\boldsymbol{A} = \begin{bmatrix} 1.2 & 2.5 & 7.6 & 0.2 \\ 5.5 & 4 & 3.3 & 9.1 \\ 8 & 0 & 4 & 1.5 \\ 4 & 5 & 6 & 7 \end{bmatrix} \quad y \quad \boldsymbol{B} = \begin{bmatrix} 1 & 2.8 & 6.6 & 0.8 \\ 5 & 4.2 & 3 & 8.8 \\ 7.6 & 0.2 & 3.9 & 1.7 \\ 4.3 & 5 & 6.3 & 7 \end{bmatrix}$$

se tiene que

$$\lambda_{\boldsymbol{A}} = [16.0110, 6.8355, -6.2365, -0.4100]$$

$$\lambda_{\boldsymbol{B}} = [16.2934, 5.5784, -5.6626, -0.1092]$$

Por otra parte, si se consideran dos matrices con elementos totalmente diferentes se tiene que

$$\boldsymbol{C} = \begin{bmatrix} 3 & 2 & -4 & 2 \\ 1 & 8 & 0 & -9 \\ 5 & 4 & 7 & 1 \\ 5 & 6 & 1 & 2 \end{bmatrix} \quad y \quad \boldsymbol{D} = \begin{bmatrix} 7 & 8 & 9 & 4 \\ 5 & -5 & 4 & 5 \\ 1 & 2 & 3 & 4 \\ 2 & -4 & 7 & 3 \end{bmatrix}$$

$$\lambda_{\boldsymbol{C}} = [6.61 + 7.47i, 6.61 - 7.47i, 1.76, 5]$$

$$\lambda_{\boldsymbol{D}} = [-5.81, -4.37, 9.09 + 2.47i, 9.09 - 2.47i]$$

Observe como efectivamente los valores propios de $\boldsymbol{A}$ y $\boldsymbol{B}$ están cercanos entre sí, mientras que los valores propios de $\boldsymbol{C}$ y $\boldsymbol{D}$ son muy diferentes entre ellos.

Un *cromagrama de entropía* se puede representar como una matriz $\boldsymbol{C}$ de tamaño $d \times n$, donde d es el número de *valores de entropía por croma* y n, el número de tramas de audio consideradas. El ADVP requiere una matriz cuadrada, por lo tanto, para calcular los valores propios de $\boldsymbol{C}$, el número de *valores de entropía por croma* debe ser igual al número de tramas de audio, es decir $d = n$.

Un SSA extrae del flujo de audio de entrada segmentos de señal de t segundos de duración en intervalos regulares de tiempo. Para el sistema propuesto se consideran segmentos de audio de un segundo de duración. Usando tramas de 100ms y un traslape entre tramas del 50 %, en un segundo se tienen 20 tramas de audio. De este modo, se

requieren 20 *valores de entropía por croma* para que el *cromagrama de entropía* sea una matriz cuadrada y se pueda aplicar el ADVP.

Uno de los objetivos de encontrar los valores propios de una matriz, es para reducir la dimensionalidad de un conjunto de datos el cual tiene un gran número de variables interrelacionadas y al mismo tiempo retener cuanto sea posible la variación presente en el conjunto de datos. A este análisis se le conoce como análisis de componentes principales. Sin embargo, no es el objetivo del método de extracción de características descrito en esta sección hacer análisis de componentes principales, sino probar la hipótesis de que *cromagramas de entropía* cuadrados de segmentos de audio con información semejante poseen valores propios similares. En caso de comprobar esta hipótesis, se estaría habilitando el uso de los valores propios como descriptores robustos en cuanto a la dinámica en amplitud que tiene una señal.

Para probar esta hipótesis, considere dos *cromagramas de entropía* cuadrados obtenidos a partir de la versión sin degradar y degradada de un segmento de audio de un segundo de duración. La degradación utilizada sobre el segmento de audio fue la adicción de ruido blanco a la señal, logrando con esto un SNR de 5dB. Ambos *cromagramas de entropía* son referidos como las matrices $\boldsymbol{A}$ y $\boldsymbol{B}$, respectivamente. El conjunto de valores propios de ambas matrices está conformado por números reales y números complejos, tal como se muestra en la Tabla 5.1.

Los datos de la Tabla 5.1 no comprueban del todo la hipótesis anterior, ya que algunos de los valores propios de $\boldsymbol{B}$ varían su valor drásticamente con respecto a los valores propios de $\boldsymbol{A}$. Sin embargo, este resultado no es del todo negativo, ya que se puede observar en esta tabla la existencia de un *Valor Propio Dominante* (VPD), es decir, que el módulo de este valor propio es mucho mayor que el módulo del resto de los valores propios, esto es, $|\lambda_1| \gg \{|\lambda_2|, |\lambda_3|, ..., |\lambda_n|\}$. Esto lleva a la siguiente cuestión: ¿Existirá siempre un VPD para todo *cromagrama de entropía* cuadrado? Si la respuesta es positiva, entonces se pueden utilizar las componentes del vector propio asociado al VPD para reducir la dimensionalidad de los datos ya que éstas representarían a una fracción dominante de los elementos del *cromagrama de entropía*. En otras palabras, los valores propios no dominantes y sus vectores propios asociados no proporcionarían ninguna información relevante acerca de los elementos del *cromagrama de entropía*, por lo tanto, se pueden descartar.

Para responder a la pregunta anterior se realizó la siguiente prueba: Dada una señal de audio completa de cualquier pieza musical, ésta se dividió en segmentos de audio de un segundo de duración sin traslape, así, se tienen tantos segmentos de audio como número de segundos tenga la pieza musical. A cada segmento de audio se le extrajo

Tabla 5.1: Valores propios de un *cromagrama de entropía* sin degradar y degradado.

Valor Propio	$\boldsymbol{A}$	$\boldsymbol{B}$
λ_1	465.3496	477.3234
λ_2	-3.7034 + 1.9429i	-3.4875 + 1.1582i
λ_3	-3.7034 - 1.9429i	-3.4875 - 1.1582i
λ_4	-0.4926 + 2.9097i	-0.1444 + 0.6849i
λ_5	-0.4926 - 2.9097i	-0.1444 - 0.6849i
λ_6	2.5237	2.1314
λ_7	1.9296 + 0.6207i	1.1062 + 0.3233i
λ_8	1.9296 - 0.6207i	1.1062 - 0.3233i
λ_9	0.2140 + 1.3890i	0.3261 + 1.1388i
λ_{10}	0.2140 - 1.3890i	0.3261 - 1.1388i
λ_{11}	-1.3296	-1.2253
λ_{12}	-0.5033 + 0.9991i	-0.6848 + 0.7502i
λ_{13}	-0.5033 - 0.9991i	-0.6848 - 0.7502i
λ_{14}	-0.8263	-1.1321
λ_{15}	-0.5621	-0.0558
λ_{16}	0.1929	0.2310
λ_{17}	0.6731 + 0.3665i	1.1929 + 1.9061i
λ_{18}	0.6731 - 0.3665i	1.1929 - 1.9061i
λ_{19}	0.8899	0.6910
λ_{20}	1.3917	0.8195

su correspondiente *cromagrama de entropía* usando las especificaciones mencionadas arriba. Finalmente, a cada *cromagrama de entropía* se le extraen sus valores propios. La gráfica del módulo de estos valores propios a través del tiempo, revelará si en todos los *cromagramas de entropía* existe un VPD.

La Figura 5.3 muestra la gráfica de los módulos de los valores propios de los *cromagramas de entropía* de tres piezas de audio diferentes. Cada gráfica de la figura muestra la existencia de un VPD que sobresale de los demás, al tener un módulo mucho mayor que el resto de los valores para todo el tiempo que dura la pieza de audio. También es importante notar como la magnitud del resto de los valores propios es muy cercana a 0 (apenas apreciable sobre el eje horizontal) para todo el tiempo. Los resultados de esta figura prueban la hipótesis anterior, la cual establece que en todo *cromagrama de entropía* cuadrado existe un único VPD.

La prueba anterior requirió calcular todos los valores propios de todos los *cromagramas de entropía* de la señal para revelar la existencia del VPD. Sin embargo, no se necesita calcular exactamente todos los valores propios para saber la magnitud de este valor propio ni tampoco su vector propio asociado. Se puede usar el método de la Iteración de la Potencia para encontrar este valor propio y su vector propio asociado. El método se basa en aplicarle a cualquier vector la transformación que representa una

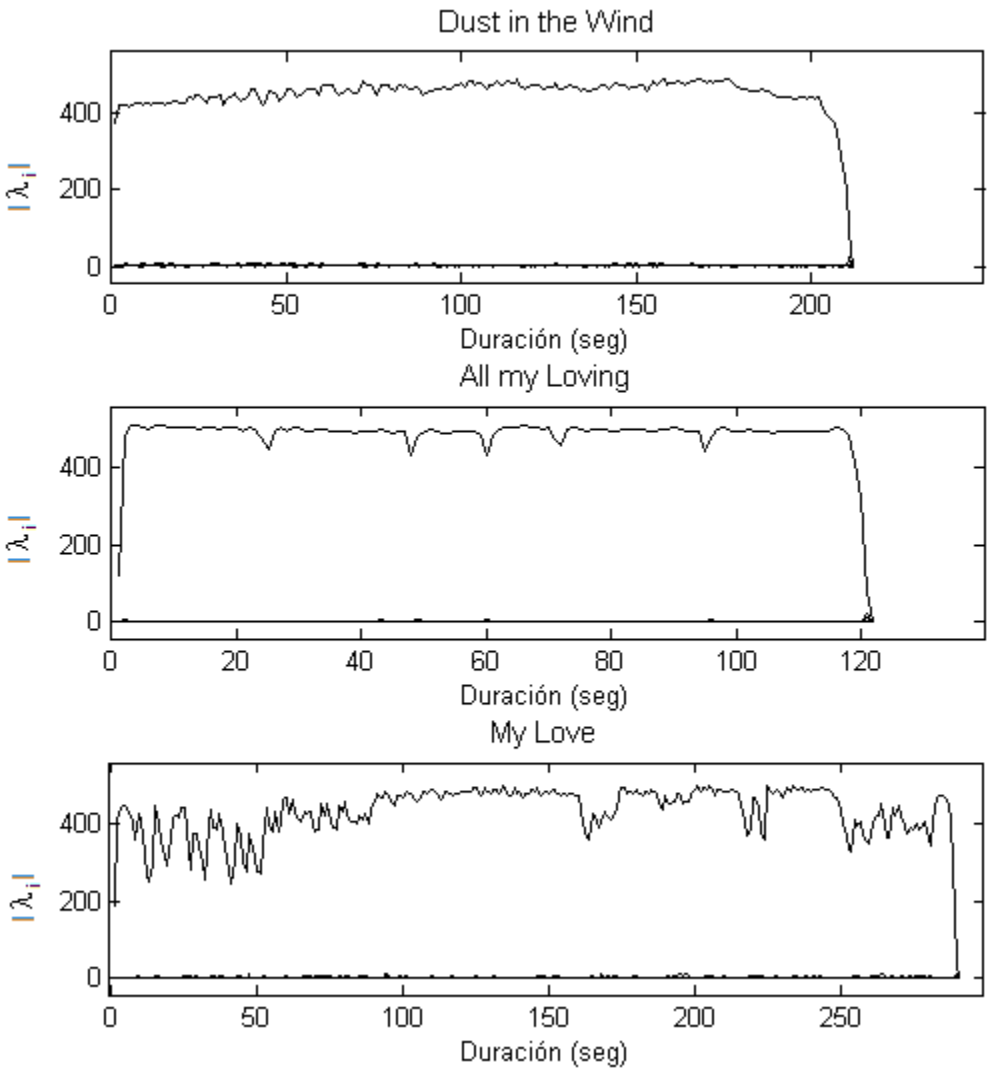

Figura 5.3: Gráficas de sucesiones de valores propios para conocer la existencia del VPD.

matriz $\boldsymbol{A}$, así, ese vector tenderá a orientarse hacia la dirección del VPD de $\boldsymbol{A}$. El objetivo del método de la Iteración de la Potencia es calcular el valor propio de mayor módulo y su vector propio asociado.

Se supone que la matriz $\boldsymbol{A}$ tiene valores propios distintos, es decir,$|\lambda_1| \neq |\lambda_2| \neq \cdots |\lambda_n|$ con $|\lambda_1| > \{|\lambda_2|, ..., |\lambda_n|\}$ y vectores propios asociados $\boldsymbol{v_1}, \boldsymbol{v_2}, ..., \boldsymbol{v_n}$. El algoritmo parte de un vector inicial $\boldsymbol{x}_0$ de módulo unidad. Éste progresa mediante una iteración de punto fijo con las ecuaciones de recurrencia dadas en (5.1) y(5.2).

$$\boldsymbol{y}_{k-1} = \boldsymbol{A}\boldsymbol{x}_{k-1} \tag{5.1}$$

$$\boldsymbol{x}_k = \frac{\boldsymbol{y}_{k-1}}{\left\|\boldsymbol{y}_{k-1}\right\|_\infty} \tag{5.2}$$

El vector $\boldsymbol{x}_i$ converge al vector característico $\boldsymbol{v_1}$, que se asocia al VPD λ_1. La magnitud de λ_1 se obtiene del valor al que converge el término $\left\|\boldsymbol{y}_{k-1}\right\|_\infty$. La velocidad de convergencia depende de la relación $\left|\frac{\lambda_2}{\lambda_1}\right|$, a menor valor de ésta, mayor velocidad de convergencia [Fuente10]. El ejemplo siguiente muestra como funciona este algoritmo. Calculemos, partiendo de $x_0^T = [\ 0 \quad 1\ ]$, el valor propio de

$$\boldsymbol{A} = \begin{pmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}$$

Utilizando el algoritmo de la Tabla 5.2, se llega al siguiente resultado:

i	$\boldsymbol{x}_i^T$	$\|\boldsymbol{x}_i\|_\infty$
0	[0.000 1.0]	
1	[0.333 1.0]	1.5000
2	[0.600 1.0]	1.6667
3	[0.778 1.0]	1.8000
4	[0.882 1.0]	1.8889
5	[0.939 1.0]	1.9412
6	[0.969 1.0]	1.9697
7	[0.984 1.0]	1.9846
8	[0.992 1.0]	1.9922
9	[0.996 1.0]	1.9961
10	[0.998 1.0]	1.9981

El comportamiento del algoritmo del método de la Potencia en cada iteración se muestra en la Figura 5.4. El vector inicial es una combinación lineal de los dos vectores propios $\boldsymbol{v_1}$ y $\boldsymbol{v_2}$. La multiplicación sucesiva por $\boldsymbol{A}$ causa que el coeficiente en el primer vector propio sea el que domine, por lo que la sucesión converge a ese vector propio.

Tabla 5.2: Algoritmo del método de la Iteración de la Potencia.

Algoritmo: Calcula el valor y vector propio dominante.
Entradas: $\boldsymbol{x}, \boldsymbol{A}$, **Salidas:** $m, \boldsymbol{x}$

Para $i = 0, ..., n$ hacer {
$\boldsymbol{v} = \boldsymbol{A}\boldsymbol{x}$
$m = max\,[abs\,(\boldsymbol{v})]$
$\boldsymbol{x} = \frac{\boldsymbol{v}}{m}$
}

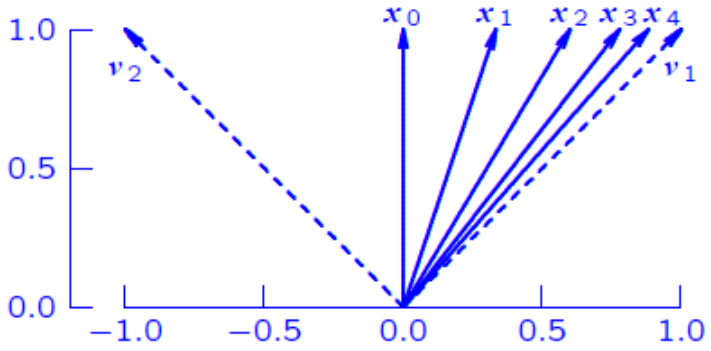

Figura 5.4: Comportamiento del algoritmo de la Iteración de la Potencia.

$$\boldsymbol{x}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 1 \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Para probar la robustez de este método con interpretaciones, se consideró el siguiente experimento. A partir de las señales de audio de dos interpretaciones diferentes de una pieza musical, se dividió cada señal en segmentos de un segundo con traslape de 100ms. Después, a cada segmento se le extrajo su correspondiente *cromagrama de entropía*. Finalmente, aplicando el método de la Iteración de la Potencia se obtuvo para cada *cromagrama de entropía* el valor y vector propio dominante. El experimento consistió en graficar la magnitud absoluta de los componentes de los vectores propios, para observar si hay una relación directa a través del tiempo entre los vectores propios de una interpretación con los vectores propios de la otra interpretación. La Figura 5.5 muestra las series de tiempo que se forman a partir del primer componente de los vectores propios de ambas interpretaciones. Observe como estas series de tiempo son muy similares en su evolución a través del tiempo, a pesar de las diferentes variaciones dinámicas de tiempo (tempo) que presentan estas dos interpretaciones. También note como la magnitud de estos componentes mantienen su proporción, lo cual quiere decir, que este método también es robusto a los cambios dinámicos de amplitud (volumen)

Tabla 5.3: Medidas de similitud entre las series de tiempo de cada componente (Medidas obtenidas con DTW).

Componente	Medida de Similitud
$\|v_1\|$	0.9985
$\|v_2\|$	0.9987
$\|v_3\|$	0.9987
$\|v_4\|$	0.9986
$\|v_5\|$	0.9988
$\|v_6\|$	0.9988
$\|v_7\|$	0.9987
$\|v_8\|$	0.9985
$\|v_9\|$	0.9987
$\|v_{10}\|$	0.9988
$\|v_{11}\|$	0.9985
$\|v_{12}\|$	0.9984
$\|v_{13}\|$	0.9979
$\|v_{14}\|$	0.9984
$\|v_{15}\|$	0.9986
$\|v_{16}\|$	0.9987
$\|v_{17}\|$	0.9985
$\|v_{18}\|$	0.9987
$\|v_{19}\|$	0.9988
$\|v_{20}\|$	0.9987

de las dos señal. Las dos interpretaciones utilizadas en el experimento de la Figura 5.5 corresponden a la pieza de audio titulada "All my Loving" interpretada por los Beatles en dos producciones diferentes.

Ahora, considere las componentes de un vector propio estar denotadas por $\boldsymbol{v} = [v_1, v_2, ..., v_n]$. De este modo, la magnitud absoluta de cada componente se puede expresar como $\{|v_1|, |v_2|, ..., |v_n|\}$. En base a lo anterior, se busca determinar que tan robusto es este método con respecto a la similitud que hay entre las series de tiempo de todas las componentes. La Tabla 5.3 muestra las medidas de similitud resultantes producto de comparar las series de tiempo que forman cada componente $|v_i|$ al graficar la sucesión de vectores propios $\{\boldsymbol{v^0}, \boldsymbol{v^1}, ..., \boldsymbol{v^t}\}$ extraídos de las dos interpretaciones diferentes de la canción "All my Loving" (el superíndice t en la sucesión se refiere al número del segmento del cual se extrajo el vector propio). Los resultados de la tabla reflejan que las series de tiempo de cada componente tiene una medida de similitud por arriba del 99 %, por lo tanto, se concluye que este método de extracción de características es robusto a cambios dinámicos de tiempo y amplitud en señales de interpretaciones diferentes de una misma pieza de audio. De este modo, para el SSA se usa el vector propio asociado al VPD como la observación que realiza el sistema para identificar la posición en el tiempo de la interpretación en vivo.

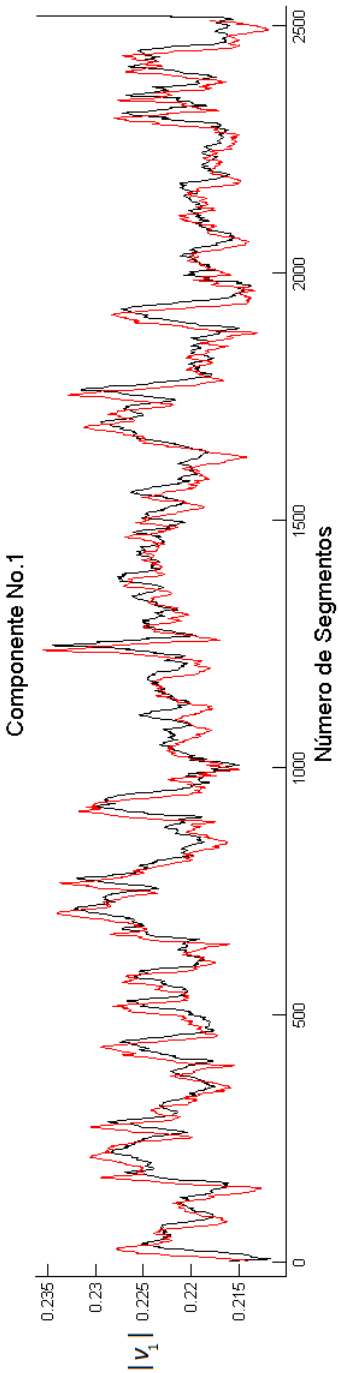

Figura 5.5: Series de tiempo generadas a partir del primer componente de los valores propios de dos interpretaciones de la canción "All my Loving".

5.3. Procesos de Markov

Las cadenas de Markov se han aplicado exitosamente en muchas aplicaciones de reconocimiento de audio. Sin embargo, los procesos de Markov en especial los procesos de decisión de Markov, sólo se usan en aplicaciones donde se tienen que tomar decisiones que condiciona la evolución futura de un sistema. En este trabajo nos centramos especialmente en revisar los procesos de decisión de Markov parcialmente observables, ya que a partir de estos procesos diseñamos la estructura interna del sistema de seguimiento de audio que interactúa con la partitura objetivo. A continuación, se proporciona una breve introducción a estos procesos.

5.3.1. Procesos de decisión de Markov

Un Proceso de Decisión de Markov (MDP por sus siglas en inglés de Markov Decision Process) es un controlador o agente que influencia el estado del sistema tomando una secuencia de acciones que optimizan la recompensa de largo plazo [Russell10, Ibe08]. Para hacer esto, el sistema observa el estado del sistema en puntos específicos de tiempo llamados "etapas de decisión" y reúne la información necesaria para elegir las acciones que hacen que el agente se desempeñe de mejor manera. Cada acción que el agente toma incurre un costo o una recompensa, y afecta el estado del sistema, de este modo, afectando acciones futuras. Así, aplicando una acción elegida al sistema, el agente incurre un costo inmediato y el sistema cambia a un nuevo estado de acuerdo a una distribución de probabilidad de transición. En general, el costo inmediato y la distribución de probabilidad de transición dependen del estado y la acción elegida. Si se denota un conjunto de etapas de decisión mediante T, entonces el proceso de decisión puede ser clasificado como un proceso de decisión de tiempo discreto o uno de tiempo continuo, dependiendo de si T es discreta o continúa. En un proceso de decisión de tiempo discreto, las decisiones son únicamente hechas en la etapa de decisión. Similarmente, en un proceso de decisión continuo, las decisiones pueden ser hechas continuamente o en puntos aleatorios cuando ciertos eventos predefinidos ocurren. En un proceso de decisión de tiempo continuo, el conjunto de etapas de decisión, T, puede ser finito o infinito. Cuando T es finito, se tiene que $T = \{1, 2, ..., N\}$, donde $N < \infty$ y los elementos de T son las etapas de decisión que están denotadas por $t \in T$. Cuando T es infinito, se tiene que $T = \{1, 2, ...\}$, que significa que las decisiones serán hechas indefinidamente. Cuando N es finita el proceso de decisión es llamado proceso de decisión de horizonte finito; de otra forma, éste es llamado proceso de decisión de horizonte infinito.

El resultado de cada decisión no es completamente predecible pero puede ser anti-

cipado a algún grado antes de que la próxima decisión sea hecha a través de la distribución de probabilidad de transición. También, las acciones aplicadas al sistema tienen una consecuencia a largo plazo debido a que las decisiones hechas en la etapa de decisión actual tienen un impacto sobre las decisiones en la etapa de decisión siguiente, y así sucesivamente. Por consiguiente, es necesario balancear lo deseado para un costo presente bajo contra la indeseabilidad de costos futuros altos. Así, buenas reglas de decisión son necesarias para especificar las acciones que deben ser tomadas en cualquier etapa de decisión y estado. Una regla para hacer decisiones en cada etapa de decisión es llamada política. Una política usada en una etapa de decisión t podría usar la historia del sistema (esto es, la secuencia de estados observados y secuencias de acciones). Sin embargo, las políticas prácticas dependen únicamente del estado observado del sistema en la etapa de decisión t. Así, una política se puede ver como una secuencia de reglas de decisión que prescribe la acción a ser tomada en todas las etapas de decisión.

Se denota una política por $D = (d_1, d_2, ..., d_{N-1})$, donde d_t es la acción a ser tomada en la etapa de decisión $t \in T$. Las políticas pueden ser clasificadas como estacionarias o no estacionarias. Una política estacionaria es aquella en la cual la misma acción a_i es tomada siempre que el sistema está en un estado dado i. Una política no estacionaria es aquella en la cual diferentes acciones pueden ser tomadas cuando el sistema está en un estado dado. La acción tomada quizá dependa sobre la etapa de decisión. Por ejemplo, para un proceso de horizonte finito, se puede tomar una acción al inicio del horizonte cuando el proceso está en el estado k y una acción diferente hacia el final del horizonte cuando el sistema está en el estado k nuevamente.

Un MDP es un sistema probabilístico de tiempo discreto que puede ser representado por la tupla (S, A, R, P), donde

- S es un conjunto finito de N estados; esto es $S = \{s_1, s_2, ..., s_N\}$. En la práctica el estado de un sistema es un conjunto de parámetros que pueden ser usados para describir el sistema.
- A es un conjunto finito de K acciones que pueden ser tomadas en cualquier estado; esto es, $A = \{a_1, a_2, ..., a_K\}$.
- R es la matriz de recompensas, cual puede variar con la acción tomada. Así, para la acción $a \in A$ se denota la recompensa asociada con una transición del estado i al estado j cuando la acción a es tomada por $r_{ij}(a)$.
- P es la matriz de probabilidad de transición, la cual puede ser diferente para cada

acción. Así, para la acción $a \in A$ se denota la probabilidad de que el proceso se mueva del estado i al estado j cuando la acción a es tomada por $p_{ij}(a)$.

Para tal sistema se tiene que

$$p(s_{n+1} = j | s_0, a_0, s_1, a_1, ..., s_n = i, a_n = a)$$

$$= p(s_{n+1} = j | s_n = i, a_n = a) = p_{ij}(a)$$

De esta manera, las probabilidades de transición y las funciones de recompensa son funciones únicas del estado anterior y la acción subsecuente. Cualquier *cadena de Markov* homogénea $\{s_n\}$ cuyas probabilidades de transición son $p_{ij}(a)$ es llamado un *proceso de decisión de Markov*, donde $\sum_j p_{ij}(a) = 1$ para todo $i \in S$ y toda $a \in A$.

5.3.2. Procesos de decisión de Markov parcialmente observables

En un MDP, la secuencia de acciones tomada para hacer decisiones asume que el ambiente es completamente observable y los efectos de las acciones tomadas son determinísticos. Esto es, en un MDP se asume que en la etapa de decisión, el estado i, las probabilidades de transición $p_{ij}(a)$, y la recompensa inmediata $r_{ij}(a)$ son todos conocidos. Sin embargo, el mundo real no es siempre completamente observable, cual significa que los efectos de las acciones tomadas son casi siempre no determinísticos. Las decisiones hechas sobre este ambiente pueden ser modeladas por un Proceso de Decisión de Markov Parcialmente Observable (POMDP por sus siglas en inglés de Partially Observable Markov Decision Process) [Smith05, Cassandra94, Kaplow10]. En un POMDP, $p_{ij}(a)$ y $r_{ij}(a)$ son todos conocidos en la etapa de decisión, pero el estado no es conocido con precisión. En este caso, el agente tiene algunas observaciones de las cuales infiere la probabilidad de que el sistema esté en algún estado. A partir de estas observaciones el agente toma una acción que resulta en una recompensa. La recompensa recibida después de que se ejecuta la acción, proporciona información de qué tan buena fue la acción que se tomó.

El agente de un POMDP elige y ejecuta una acción en la etapa de decisión basándose en la información de las observaciones pasadas, acciones pasadas y la observación actual. Desafortunadamente, la cantidad de memoria requerida para almacenar las observaciones y las acciones pasadas puede ser grande, de este modo, es difícil mantener la información pasada después de un periodo largo de tiempo. Esta dificultad es usualmen-

te sobrellevada manteniendo el *estado de creencia* del agente en lugar de la información pasada, donde un *estado de creencia* es la distribución de probabilidad sobre los estados ocultos del proceso de Markov dada la historia pasada de las observaciones y acciones. Así, el *estado de creencia* captura toda la información pasada y la observación actual que es útil para seleccionar una acción. Debido a que el número de posibles estados del ambiente es finito, mantener el *estado de creencia* es más simple que mantener la trayectoria de toda la información pasada. Note que el hecho de que el *estado de creencia* esté definido en términos de una distribución de probabilidad implica que el conocimiento del agente está incompleto. También, usando el concepto de *estado de creencia* permite al POMDP satisfacer la propiedad de Markov debido a que si se conoce el *estado de creencia* actual se puede predecir el futuro. Cuando el agente observa el estado actual del ambiente este actualiza su *estado de creencia*.

En un POMDP el agente tiene que resolver dos problemas simultáneamente, es decir, un problema de control como ocurre en un MDP y un problema de identificación para los estados no observados. En cada tiempo que el agente toma una acción, la transición a un nuevo estado implícitamente proporciona nueva información acerca del estado del proceso. Este nuevo conocimiento puede habilitar al agente a hacer la próxima decisión. Así, empezando con una distribución de probabilidad inicial, el agente revisa la distribución después de cada transición para tomar en consideración la nueva información proporcionada por la observación resultante de la transición. A esta distribución se le conoce como distribución a posteriori, que es usada para identificar el estado no observado y controlar el sistema en la próxima etapa de decisión. Mas formalmente, un POMDP es un modelo probabilístico que puede ser representado por la tupla $(S, A, \Omega, P, \Psi, R)$, donde

- S es un conjunto finito de N estados del núcleo del proceso; esto es, $S = \{s_1, s_2, ..., s_N\}$. El estado en el tiempo t es denotado por S_t.
- A es un conjunto finito de K acciones que pueden ser tomadas en cualquier estado; esto es, $A = \{a_1, a_2, ..., a_K\}$. La acción tomada en el tiempo t es denotada por A_t.
- Ω es un conjunto finito de M observaciones que pueden ser hechas; esto es, $\Omega = \{o_1, o_2, ..., o_M\}$. La observación en el tiempo t es denotada por Ω_t.
- P es la matriz de probabilidad de transición, que puede ser diferente para cada acción. Como en el caso de un MDP, para la acción $a \in A$ se denota la probabilidad de que el sistema se pase del estado s_i al estado s_j cuando la acción a es tomada,

por $p_{ij}(a)$, cual es independiente de la historia del proceso hasta el tiempo en que la acción fue tomada. Esto es,

$$p_{ij}(a) = p(S_{t+1} = s_j | S_t = s_i, A_t = a)$$

Como se mencionó anteriormente, es asumido que los $p_{ij}(a)$ son conocidos, pero el estado s_i no es conocido en la etapa de decisión; éste es inferido a partir de la observación.

- Ψ es el conjunto de probabilidades de observación que describe la relación entre las observaciones, estados del núcleo del proceso y acciones. Sea $\varphi_{ij}(a)$ la probabilidad de observar $o_j \in \Omega$ después de que la acción a es tomada y el núcleo del proceso entra al estado s_i. Esto es,

$$\psi_{ij}(a) = p(\Omega_t = o_j | S_t = s_i, A_{t-1} = a)$$

- R es la función de recompensa, cual puede variar con la acción tomada. La recompensa en el tiempo t es denotada por R_t. La recompensa que el agente recibe tomando la acción $a \in A$ en el estado s_i que resulta en una transición al estado s_j es denotado por $r_{ij}(a)$. La recompensa total asociada con la acción a en el estado s_i es

$$r_i(a) = \sum_{s_j \in S} r_{ij}(a) p_{ij}(a)$$

Asuma que el núcleo del proceso está en el estado S_t en el tiempo t. Debido a que un POMDP está basado en un núcleo el cual es un proceso de Markov, el estado actual S_t es suficiente para predecir el futuro independientemente de los estados pasados $\{S_0, S_1, ..., S_{t-1}\}$. El estado S_t no es directamente observable pero puede ser inferido a partir de las observaciones $\{\Omega_1, \Omega_2, ..., \Omega_{t-1}\}$. Para ayudar a determinar el estado del sistema, el agente mantiene un trazo completo de todas las observaciones y todas las acciones que éste ha tomado y usa esta información para elegir las siguientes acciones. El trazo conjunto de acciones y observaciones constituye una historia al tiempo t, cual es denotada por H y definida por

$$H_t = \{A_0, \Omega_1, A_1, \Omega_2, ..., A_{t-1}, \Omega_t\}$$

Afortunadamente, esta historia no necesita ser representada explícitamente y puede

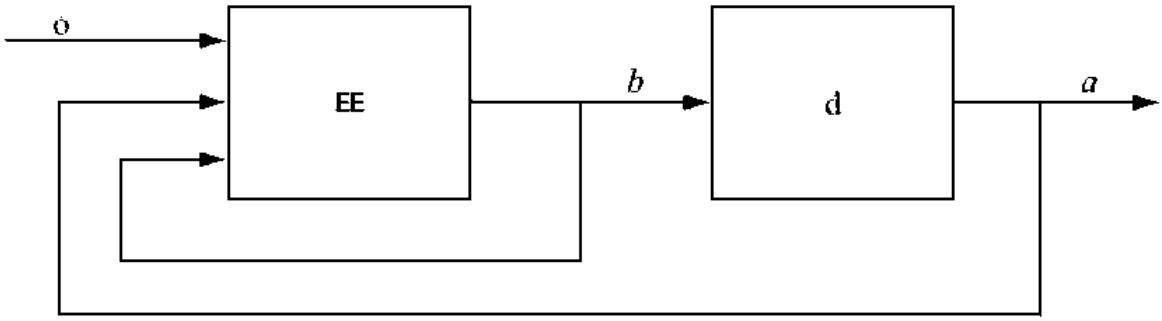

Figura 5.6: Estructura del proceso de estimación de estado [Ibe08].

ser resumida vía una distribución de creencia $b_t(s)$, cual está definida por

$$b_t(s) = p(S_t = s|\Omega_t, A_{t-1}, ..., \Omega_1, A_0, b_0)$$

De esta manera, $0 \leq b_t(s) \leq 1$ es la probabilidad de que el proceso esté en el estado $S_t = s$ dada la distribución de creencia b. Esto es, $b_t(s_j)$ es la estimación del agente de que el núcleo del proceso se encuentre en el estado $S_t = s_j$. Por lo tanto, basándose en el *estado de creencia* actual el agente elige una acción a y recibe la recompensa $r_{jk}(a)$, y el núcleo del proceso hace una transición al estado s_k que conduce a la observación o_m. Esto es ilustrado en la Figura 5.6 donde el componente etiquetado como "EE" es el estimador de estado que toma como entrada el último *estado de creencia*, la más reciente acción, y la más reciente observación y regresa un *estado de creencia* actualizado [Ibe08]. El componente etiquetado por "d" representa la política.

La distribución de probabilidad de estado inicial, cual define la probabilidad de que el sistema esté en el estado s en el tiempo $t = 0$, está dada por

$$b_0(s) = p(S_0 = s)$$

Los métodos usados para resolver POMDP son algunas veces llamados algoritmos de aprendizaje por reforzamiento debido a que la única retroalimentación proporcionada al agente es una señal de recompensa escalar en cada paso. Una característica importante de b_t es el hecho de que puede ser determinado recursivamente usando únicamente el valor pasado inmediato, b_{t-1}, junto con la más reciente acción A_{t-1} y la observación Ω_t. Si se denota el *estado de creencia* para el estado $S_t = s_k$ en el tiempo t por $b_t(s_k)$, entonces basado en A_{t-1} y Ω_t, la distribución de creencia es actualizada vía la siguiente

regla de Bayes:

$$b_t(s_k) = p(S_t = s_k|\Omega_t = o_m, A_{t-1} = a, ..., \Omega_1, A_0) = p(s_k|o_m, a, b_{t-1}(s_j))$$

$$= \frac{p(s_k, o_m, a, b_{t-1}(s_j))}{p(o_m, a, b_{t-1}(s_j))} = \frac{p(o_m|s_k, a, b_{t-1}(s_j))\, p(s_k, a, b_{t-1}(s_j))}{p(o_m|a, b_{t-1}(s_j))\, p(a, b_{t-1}(s_j))}$$

$$= \frac{p(o_m|s_k, a, b_{t-1}(s_j))\, p(s_k|a, b_{t-1}(s_j))\, p(a, b_{t-1}(s_j))}{p(o_m|a, b_{t-1}(s_j))\, p(a, b_{t-1}(s_j))}$$

$$= \frac{p(o_m|s_k, a, b_{t-1}(s_j))\, p(s_k|a, b_{t-1}(s_j))}{p(o_m|a, b_{t-1}(s_j))}$$

$$= \frac{p(o_m|s_k, a) \sum_{s\in S} p(s_k|a, b_{t-1}(s), s)\, p(s|a, b_{t-1}(s))}{p(o_m|a, b_{t-1}(s_j))}$$

$$= \frac{p(o_m|s_k, a) \sum_{s\in S} p(s_k|a, s_j)\, b_{t-1}(s_j)}{p(o_m|a, b_{t-1}(s_j))}$$

$$= \frac{\varphi_{km}(a) \sum_{s_j\in S} p_{jk}(a)\, b_{t-1}(s_j)}{p(o_m|a, b_{t-1}(s_j))} \tag{5.3}$$

El denominador es independiente de s_k y puede ser considerado como un factor de normalización. El numerador contiene la función observación, la probabilidad de transición y el *estado de creencia* actual. Así se tiene que

$$b_t(s_k) = \rho\varphi_{km}(a) \sum_{s_j\in S} p_{jk}(a)\, b_{t-1}(s_j) \tag{5.4}$$

donde ρ es una constante de normalización. Debido a que el *estado de creencia* b_t en el tiempo t es determinado recursivamente usando b_{t-1}, así como la más reciente observación Ω_t y la más reciente acción A_{t-1}, se puede definir la actualización del *estado de creencia* por la siguiente operación;

$$b_t(s) = \tau(b_{t-1}, A_{t-1}, \Omega_t)$$

donde $\tau(b_{t-1}, A_{t-1}, \Omega_t)$ se le conoce como función de actualización de creencia. Esto muestra que dado un *estado de creencia*, su *estado de creencia* sucesor es determinado

por la acción y observación.

5.3.3. Proceso de Markov parcialmente observable para alineamiento de audio

Los POMDP se aplican extensivamente para planear en ambientes donde el conocimiento de un proceso es confundido por factores desconocidos. Aunque esta característica propia de ellos los hace una herramienta potencial, su solución exacta es intratable en tareas de decisión secuencial o a largo plazo. Sin embargo, se han desarrollado soluciones aproximadas que permiten utilizarlos en dichas aplicaciones. En esta sección se introduce un nuevo proceso de Markov (propuesto por nosotros) el cual opera bajo condiciones parcialmente observables y que es usado de núcleo del sistema de seguimiento de audio para que interactué con la partitura objetivo. Este proceso se basa en un modelo estocástico que al igual que los POMDP mantienen el *estado de creencia* para la toma de desiciones. La ventaja que tiene este modelo es que no se necesita solucionar el problema de elegir las acciones óptimas en cada paso del tiempo o a largo plazo. Al existir únicamente el *estado de creencia* dentro del modelo su complejidad matemática disminuye, reduciendo el problema de los POMDP a un problema de seguimiento de la trayectoria del modelo a través del espacio de creencia. La simplicidad matemática en el modelo propuesto hace que se pueda comparar contra los HMM para así evaluar su desempeño.

Un POMDP es un modelo probabilístico que puede ser representado por la tupla $(S, A, \Omega, P,\ \Psi, R)$ descrita en la sección anterior. Este modelo asume observabilidad parcial, es decir, no existe un mapeo uno a uno de obervaciones a estados. De esta manera, el modelo puede sufrir alias perceptual en el cual el estado actual no puede ser definitivamente identificado usando los datos disponibles. En consecuencia, un POMDP mantiene una distribución de probabilidad continua sobre S, el *estado de creencia*. El conjunto de todos los posibles *estados de creencia* forman el espacio de creencia. La ecuación (5.4) es usada para actualizar el valor del *estado de creencia* $b_t(s)$.

A diferencia de los POMDP convencionales, la elección de la acción pierde su relevancia en este modelo. El objetivo ya no es elegir la acción óptima, sino seguir la trayectoria del modelo a través del espacio de creencia. Una vez que la "elección" es removida de un POMDP, el conjunto de acciones colapsa. Para los propósitos del modelo propuesto, se asume que existe dos condiciones, cambiar el estado y mantener el estado. De esta manera, la ecuación (5.4) se reduce a la ecuación (5.5),

Tabla 5.4: Algoritmo para obtener la trayectoria de creencia seguida por el modelo.

Algoritmo: Calcula el nuevo *estado de creencia* para generar la trayectoria del modelo	
Entrada: $b_0(s) \leftarrow$Asignar estado inicial	**Salida:** $b_t(s_k)$ y $Path_t$
$t = 0$. Para $\forall o_m \in \Omega$ hacer { Para $\forall s_k \in S$ $b_t(s_k) = \rho\varphi_{km} \sum_{s_j \in S} p_{jk} b_{t-1}(s_j)$ end $Path_t = max\{b_t(s_k)\}$ $t = t + 1$ end	

$$b_t(s_k) = \rho\varphi_{km} \sum_{s_j \in S} p_{jk} b_{t-1}(s_j) \tag{5.5}$$

donde φ_{km} es la probabilidad de observar $o_m \in \Omega$ y que el núcleo del proceso entre al estado s_k. Iterando (5.5) sobre todos los posibles estados se consigue la trayectoria de creencia seguida por el modelo a través del tiempo. De aquí en adelante nos referiremos a este tipo de modelo como *Proceso de Markov Parcialmente Observable* (PMPO). El algoritmo general de para determinar la trayectoria del PMPO a través del espacio de creencia es dado en la Tabla 5.4.

En muchos aspectos, la arquitectura del PMPO se asemeja a la de un HMM. Ambos esquemas comprenden un conjunto finito de estados, una matriz dictando las probabilidades de transición y una matriz designando las densidades de probabilidad observacional. Con la retirada de las acciones en el modelo clásico del POMDP, esta similitud incrementa, ya que la noción de acción y recompensa no tienen validez dentro del PMPO. Además, el PMPO no tiene la necesidad que tiene un HMM sobre el vector de probabilidad de estado inicial. El PMPO es asumido a empezar en un estado inicial designado con completa certidumbre.

La topología de un PMPO puede ser del tipo ergódico o de izquierda – derecha en la práctica. En la fase de entrenamiento, al PMPO se le proporciona una secuencia de observaciones de entrenamiento (vectores propios en nuestro caso) para estimar la matriz de transición P y las distribuciones o funciones de densidad de probabilidad de las observaciones. Otra de las ventajas que proporciona la similitud entre los PMPO y los HMM, es que se puede utilizar el algoritmo de Baum Welch para estimar los parámetros del modelo que maximizan la probabilidad de generación de dicha secuencia.

5.4. Experimentos y Resultados

En esta sección se proporcionan los experimentos y resultados preliminares que son obtenidos al considerar los vectores propios de cromagramas de entropía en un SSA. En estos experimentos los vectores propios son utilizados para representar la partitura objetivo (audio de referencia) y las observaciones del sistema (audio a seguir). El objetivo de los experimentos es contar el número de estados que son estimados correctamente mediante un PMPO tomando como base la ruta óptima de la secuencia de estados generada fuera de línea con el algoritmo de Viterbi. En principio, se explican las consideraciones a tomar en cuenta para obtener los vectores propios de la partitura objetivo y las observaciones del SSA. Posteriormente, se dedica una sección para describir el proceso para determinar el modelo del SSA y finalmente, se termina con una sección dedicada la prueba del modelo del SSA usando 8 pares de interpretaciones.

5.4.1. Pre-procesamiento de la señal

El experimento preliminar de esta sección se hizo utilizando 8 pares de interpretaciones de una pieza musical. Cada par de interpretaciones corresponde a las señales de audio de una pieza musical tocada con diferente instrumentación y ejecutada en diferente momento. El género de música de estas piezas son classical crossover, rock, pop y música clásica. Todas las piezas de audio están en formato WAV, tienen codificación PCM a 16 bits y están muestreadas a 44100Hz. Estas interpretaciones pertenecen a una colección personal por lo que no se encuentran disponibles en ninguna base de datos o servidor, sin embargo, no se cometió ninguna infracción de derechos de autor.

Para poder usar estas interpretaciones en el SSA, fue necesario aplicar un paso de pre-procesamiento a las señales de audio de cada una de ellas. Este paso se realiza siguiendo el siguiente procedimiento:

- De cada par de interpretaciones, la señal de audio de una de ellas se considera la partitura objetivo (señal de referencia) respecto a la cual se hará el seguimiento, mientras que la otra representa la interpretación a adquirir en vivo.
- La señal de audio utilizada de partitura objetivo es dividida en tramas de 100ms con un traslape entre ellas del 50 % (Esto asegura que se consideren pocos eventos perceptuales en una trama y además entre trama y trama exista una correlación temporal).

- A cada trama se le extraen 20 *valores de entropía por croma*, por lo tanto, por cada trama se tiene un vector 20-dimensional.
- Una matriz de tamaño 20×20 es formada considerando una sucesión de 20 vectores consecutivos. Usando este procedimiento se forman tantas matrices como permita la señal, utilizando un paso de una trama de avance entre ellas.
- A cada matriz se le extrae el vector propio dominante utilizando el método de la Iteración de la Potencia.
- Se aplica el valor absoluto a cada componente de los vectores propios.
- Respetando el orden en el que fueron calculados los componentes de estos vectores, se forman secuencias de datos, es decir, el comportamiento del componente v_i en el tiempo describe una secuencia de datos, tal como fue mostrado en la Figura 5.5.
- Las secuencias de datos que corresponden a la partitura objetivo son almacenadas en una base de datos para su posterior uso.
- Finalmente, la señal de audio que representa a la interpretación en vivo, es procesada de la misma manera, con la única diferencia de que entre cada matriz hay 20 tramas de separación, es decir, cada segundo de audio se extrae una matriz.

5.4.2. Obtención del modelo

El modelo propuesto considera distribuciones de probabilidad discretas sobre el conjunto de observaciones. Esto implica que en el proceso de reconocimiento se realice una cuantización escalar de los posibles valores de las componentes de los vectores propios observados.

En el experimento preliminar de esta sección, se utiliza únicamente una de todas las secuencias de datos para determinar el conjunto de símbolos observables que puede producir la señal de audio a seguir. Para cuantizar esta secuencia de datos continuos, se utiliza el modelo de mezcla de gaussianas. El método de cuantización consiste en determinar los pesos, medias y varianzas de los k componentes gaussianos que mejor ajustan a la distribución de los datos de la secuencia. Así, se tienen tantos símbolos como componentes gaussianos tenga la función de densidad de probabilidad. Los símbolos utilizados son referenciados con el conjunto de números enteros positivos, esto es, $\Omega = \{1, 2, ...\}$. Cuando el SSA realiza una observación, éste la evalúa sobre cada componente gaussiano. El símbolo al que pertenece la observación corresponde al número de componente gaussiano que emita la mayor probabilidad de que dicha observación

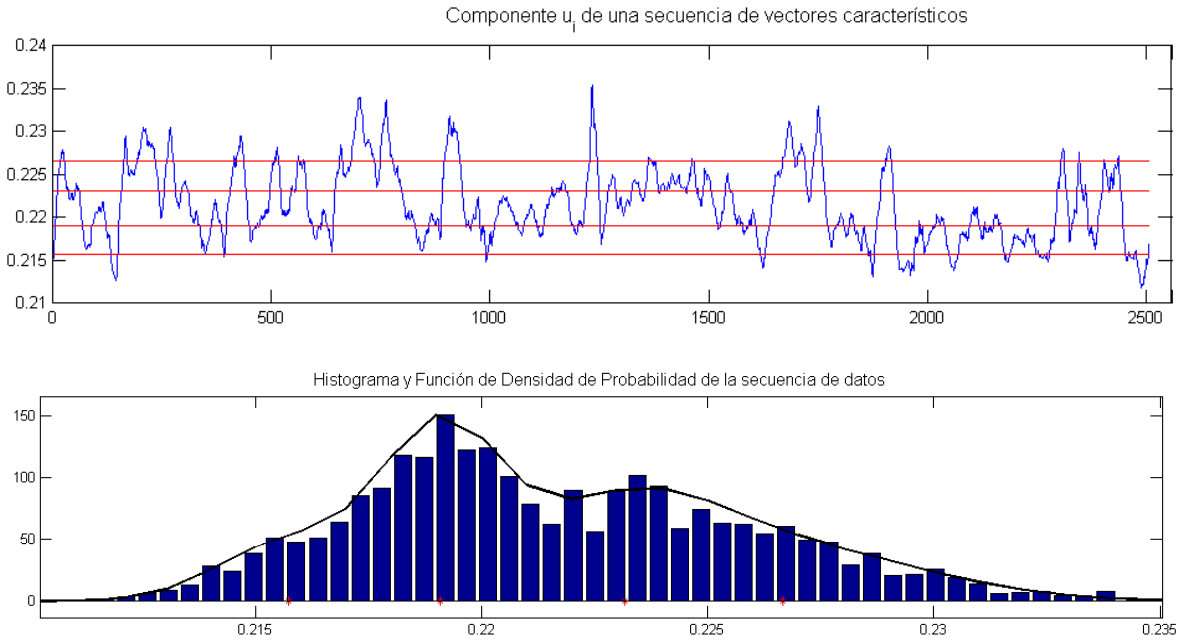

Figura 5.7: Método de cuantización escalar usando el modelo de mezcla de gaussianas.

pertenezca a ese componente. La Figura 5.7 muestra un ejemplo de este método de cuantización el cual utiliza 4 componentes gaussianos. En la parte superior de la figura se muestra la gráfica del comportamiento en el tiempo del componente v_i de una secuencia de vectores propios. Sobre esta gráfica aparecen dibujadas líneas horizontales que pertenecen a las medias de cada componente gaussiano. En la parte inferior de la figura se muestra el histograma de los datos de la secuencia y la gráfica de la función de densidad de probabilidad continua que los ajusta.

La Figura 5.8 muestra la secuencia generada de símbolos producto del proceso de cuantización de la secuencia de datos de la Figura 5.7.

El modelo puede ser del tipo ergódico o de izquierda – derecha. En este experimento preliminar el modelo usado fue de tipo ergódico (dejamos para trabajo futuro la prueba con modelos de izquierda-derecha), es decir, todas las transiciones entre estados del sistema son posibles. El modelo está definido por una matriz de probabilidad de transición entre estados, P, y un conjunto de probabilidades de observación por cada estado, Ψ, el cual puede ser representado mediante una matriz de probabilidades de observación, con elementos φ_{ij}. Un estimado inicial de la matriz Ψ puede ser obtenido a partir de una secuencia de símbolos. Este estimado consiste en determinar la frecuencia de los símbolos o_k. Esto forma un histograma con el cual se puede determinar la probabilidad con la que el símbolo o_k aparece en la secuencia, por lo tanto, debe cumplirse que

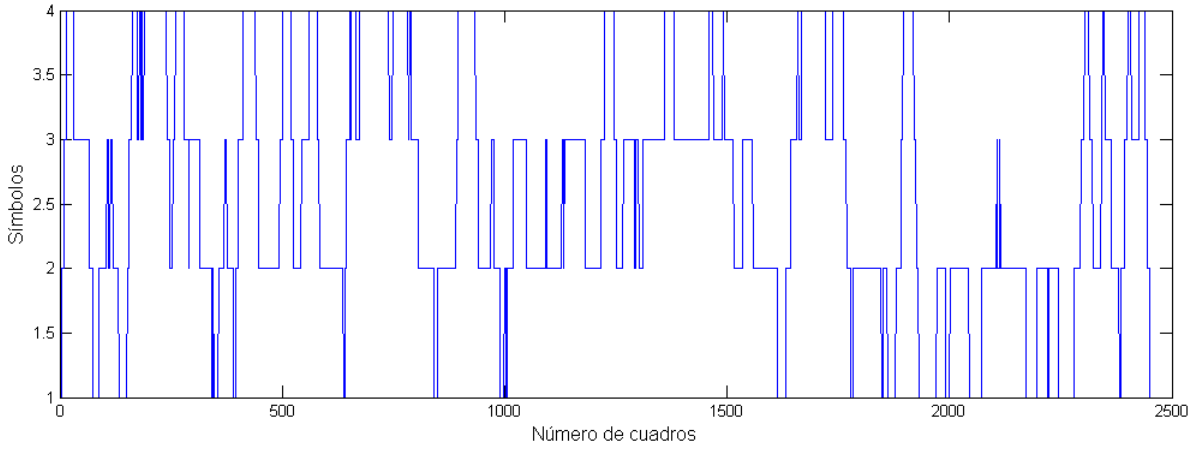

Figura 5.8: Secuencia de símbolos generada a partir del proceso de cuantización.

$$\sum_{j=1}^{k} \varphi_{ij} = 1 \quad 1 \leq i \leq N$$

El estimado inicial de Ψ asume que todos los estados i (renglones de la matriz) tienen la misma probabilidad de observar los símbolos o_k, esto es,

$$\Psi = \begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1k} \\ \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1k} \\ \vdots & \vdots & \vdots & \vdots \\ \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1k} \end{bmatrix}$$

La matriz de transición de probabilidades P es inicializada con probabilidades aleatorias, cumpliendo que

$$\sum_{j=1}^{N} p_{ij} = 1 \quad 1 \leq i \leq N$$

donde N es el número de estados. Para encontrar los parámetros óptimos del modelo $\lambda = (P, \Psi, \Pi)$, se utiliza el algoritmo de Baum Welch. El conjunto de probabilidades de estado inicial Π, es inicializado con una probabilidad de $1/N$ para los N estados del modelo. Así, el modelo queda entrenado para que la probabilidad de generación de la secuencia sea óptima.

5.4.3. Prueba del modelo

Para la prueba del modelo considere de ejemplo la secuencia de símbolos mostrada en la Figura 5.8. Esta secuencia de símbolos consta de 4 símbolos diferentes, esto es, $\Omega = \{1, 2, 3, 4\}$. Para este ejemplo se utiliza un modelo ergódico de 4 estados. Los parámetros iniciales antes de entrenar el modelo fueron los siguientes:

$$\Pi = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}$$

$$P = \begin{bmatrix} 0.2 & 0.1 & 0.3 & 0.4 \\ 0.5 & 0.2 & 0.15 & 0.15 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.2 & 0.3 & 0.3 & 0.2 \end{bmatrix}$$

$$\Psi = \begin{bmatrix} 0.075 & 0.3916 & 0.35 & 0.1833 \\ 0.075 & 0.3916 & 0.35 & 0.1833 \\ 0.075 & 0.3916 & 0.35 & 0.1833 \\ 0.075 & 0.3916 & 0.35 & 0.1833 \end{bmatrix}$$

Después de entrenar el modelo por medio del algoritmo de Baum Welch, los parámetros estimados del modelo fueron los siguientes:

$$\Pi = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$$

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0.204412054397831 & 0.795587945602169 & 0 & 0 \\ 0.107809182348805 & 0 & 0.495402081304 & 0.396788736346 \\ 0 & 0.452940360327308 & 0.547059639672 & 0 \end{bmatrix}$$

$$\Psi = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0.163269948927757 & 0.796828721436138 & 0 & 0.039901329636 \\ 0 & 0 & 0.423410421306234 & 0.576589578693 \\ 0 & 0.124870621976721 & 0.689888039285266 & 0.185241338738 \end{bmatrix}$$

Para probar el modelo se utiliza la secuencia de datos de la otra versión de la interpretación de la pieza musical. Esta secuencia tiene un tamaño igual al número de

segundos que dura la interpretación. Nuevamente se usa el método de cuantización para generar la secuencia de observaciones que el SSA realiza durante la ejecución de la interpretación en vivo. En la Figura 5.9 se muestra la secuencia de símbolos generada por la partitura objetivo (línea roja) y la secuencia de observaciones generada por la interpretación en vivo (línea negra). Observe como hay una gran similitud entre ambas secuencias.

Las dos secuencias anteriores pueden usarse para determinar el alineamiento entre las señales de audio de las dos interpretaciones (este experimento se ha dejado para trabajo futuro). El objetivo de esta prueba es obtener fuera de línea la decodificación de la secuencia de estados más probable dada la secuencia de observaciones y el modelo, para demostrar que dicha secuencia de estados puede ser generada también a través del *estado de creencia* $b_t(s_k)$. Para obtener la decodificación de la secuencia de estados más probable se evalúa el algoritmo de Viterbi. Por otra parte, para obtener la trayectoria del modelo a través del espacio de creencia se evalúa el algoritmo dado en la Tabla 5.4. En la Figura 5.10 se muestran las secuencias de estados más probables proporcionadas por ambos algoritmos. La línea roja corresponde a la secuencia de estados generada por el algoritmo de Viterbi, mientras que la línea negra corresponde a la secuencia de estados generada por el *estado de creencia*. El resultado de la figura revela que el *estado de creencia* es un buen estimador de la secuencia de estados que es generada por el algoritmo de Viterbi, ya que como se observa en la Figura 5.10 únicamente erró en muy pocas ocasiones.

Sin embargo, la ventaja que se tiene al usar el *estado de creencia* en cada paso del tiempo, es que la secuencia de estados más probable se está generando conforme el sistema realiza una observación. Esta característica del *estado de creencia* habilita al modelo para que pueda hacer alineamiento de audio en tiempo real. La Tabla 5.5 proporciona los resultados de la comparación entre las secuencias generadas por el algoritmo de Viterbi y el algoritmo para obtener la trayectoria de creencia. La primera columna de la tabla muestra los nombres de 8 interpretaciones diferentes. La segunda y tercera columna indica si el experimento fue llevado a cabo comparando interpretaciones grabadas en estudio, en vivo o ambas. La cuarta columna proporciona la duración promedio de las canciones. Finalmente, la quinta y sexta columna muestran el número de estados mal estimados y el porcentaje de error, respectivamente. Es claro observar como el porcentaje de error está por debajo del 20 %, lo que significa que el algoritmo para obtener la trayectoria de creencia se desempeña muy bien para hacer alineamiento de audio en tiempo real. Para este experimento se usaron PMPO de tipo ergódico de 4 estados y 4 símbolos (el número de estados y símbolos se obtuvo en base a la experimen-

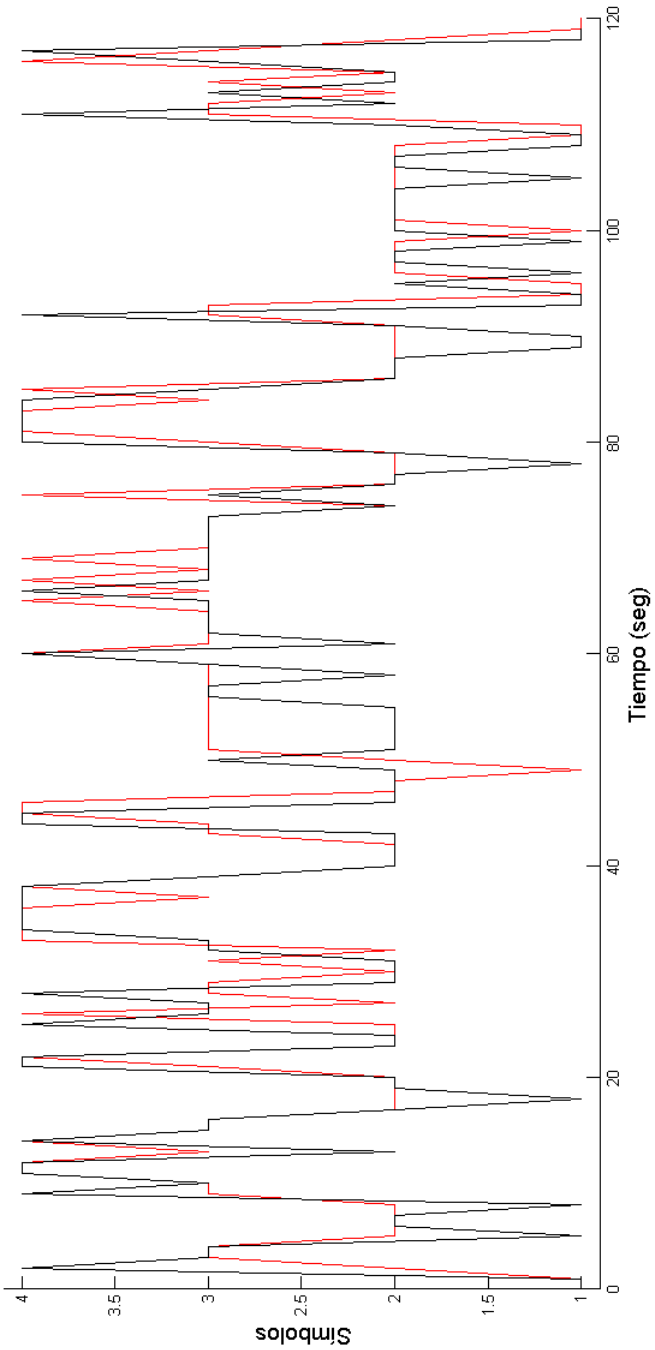

Figura 5.9: Secuencias de símbolos generadas a partir de la partitura objetivo y la interpretación en vivo.

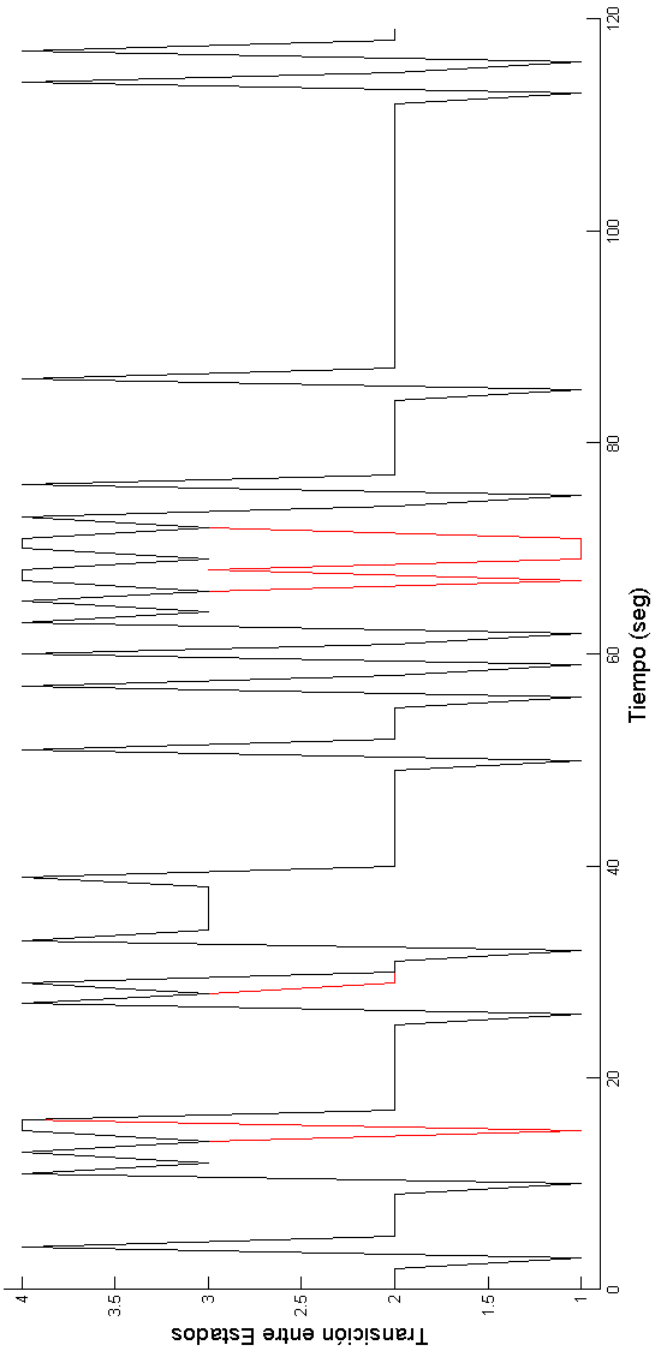

Figura 5.10: Secuencias de estados más probables obtenidas por el algoritmo de Viterbi y por el algoritmo de la Tabla 5.4.

Tabla 5.5: Resultados sobre estimación de estados usando los algoritmos de Viterbi y estado de creencia.

Pieza de Audio	Estudio	Vivo	Seg.	Errores	% de Error
Yellow submarine (Beatles)	x		144	14	9.72
Dust in the wind (Sarah Brightman)	x	x	209	38	18.18
Mo Domine (Orquestas diferentes)	x		212	31	14.62
Hurts like heaven (Coldplay)	x	x	227	25	11.01
Puedes contar conmigo (La oreja de van gogh)	x	x	220	19	8.63
Iam addicted (Madonna)	x	x	265	28	10.56
All by myself (Celine Dion)	x	x	288	8	2.7
Geografía	x	x	184	16	8.69

tación, por lo que este tema está abierto a futuras discusiones). En la siguiente sección se extienden los comentarios sobre estos resultados y se proporcionan las conclusiones.

5.5. Conclusiones y Comentarios

En este capítulo se introdujo un nuevo método para hacer seguimiento de audio. Este método se dividió en dos partes principales; el proceso de extracción de características de la señal de audio y el modelo del sistema de seguimiento de audio.

El proceso de extracción de características se llevó a cabo en dos partes: La primera parte consistió en obtener los *cromagramas de entropía* de una señal de audio. Los *cromagramas de entropía* fueron elegidos debido a que son robustos a distorsiones y variaciones de dinámica en las señales de audio. La segunda parte del proceso involucró un análisis propuesto por nosotros al cual se le denominó análisis de descomposición de valores propios. Con este análisis se pudo determinar la existencia de un valor característico dominante que tiene módulo mucho mayor que el resto de los valores propios de un *cromagrama de entropía*. Las matrices utilizadas en este análisis corresponden a *cromagramas de entropía* cuadrados. Para determinar la magnitud del VPD se utilizó el método de la Iteración de la Potencia el cual siempre converge debido a que siempre existe este valor. Con el método de la Iteración de la Potencia también se puede encontrar el vector propio asociado al VPD.

A partir del análisis de diferentes sistemas se sabe que las componentes del vector propio asociado al VPD representan a una fracción dominante de los elementos de la matriz que representa al sistema. Dado que aquí el resto de los valores propios tienen un módulo muy pequeño respecto al módulo del VPD, la información que nos proporcionan las componentes de sus vectores asociados no es relevante, por lo tanto, fueron descartados. Así, el proceso de extracción de características consistió en encontrar los vectores propios dominantes para los *cromagramas de entropía* de una señal de audio.

Por otra parte, la gráfica de los módulos de las componentes de estos vectores forman series de tiempo que son muy robustas a las variaciones dinámicas de tiempo y amplitud de señales de interpretaciones.

Para el modelo del SSA se usó un *proceso de Markov parcialmente observable*. Para poder aplicar este proceso al problema de seguimiento de audio, se necesitó considerar el concepto de *estado de creencia*. Este concepto permite disminuir la complejidad matemática del modelo para seguir la trayectoria del modelo a través del espacio de creencia. Para determinar los parámetros óptimos del modelo se usaron secuencias de vectores propios. Utilizando estas secuencias y el modelo de mezcla de gaussianas como método de cuantización escalar, se generaron las secuencias de símbolos para entrenar el modelo. El modelo puede ser entrenado usando el algoritmo de Baum Welch debido a la semejanza que tienen con los HMM.

En el experimento se trabajó con pares de interpretaciones de una pieza musical, donde la señal de audio de una de las interpretaciones fue usada de partitura objetivo, mientras que la otra se utilizó como la señal que adquiere el sistema en vivo.

El experimento preliminar de esta contribución consistió en probar que las secuencias de símbolos generadas a partir de la partitura objetivo y la interpretación en vivo, generan la misma trayectoria de estados usando tanto el algoritmo de Viterbi como la trayectoria del modelo a través del espacio de creencia. Tal comprobación fue mostrada en la Figura 5.10 donde es claro observar como a lo largo del tiempo se estiman en su mayoría los mismos estados por ambos métodos. En experimentos futuros se probará que mediante el uso de las secuencias de todos los componentes, este estimado sea completamente el mismo por ambos métodos. Con respecto a la Tabla 5.5, en especial haciendo referencia a la columna que muestra el porcentaje de error, creemos que involucrando más componentes permitirá que este porcentaje disminuya, en otra palabras, si se incrementa la dimensionalidad al problema, el modelo del sistema estimará exactamente los mismos estados que con el algoritmo de Viterbi. De cualquier forma, estos resultados nos motivan ya que el sistema fue capaz de estimar la mayoría de los estados a pesar de usar piezas de audio grabadas en estudio y piezas de audio grabadas en vivo donde diferentes factores distorsionan la señal de audio.

Capítulo 6

CONCLUSIONES Y TRABAJO FUTURO

6.1. Conclusiones Generales

En este trabajo se presentaron tres aportaciones al estado del arte, todas ellas relacionadas con los problemas de caracterización y reconocimiento de audio. La caracterización de audio es un problema que generalmente viene acompañado con el tópico de robustez, donde juntos tienen el objetivo de representar adecuadamente los datos de audio, aún si están corrompidos, para que un sistema los pueda clasificar y reconocer. En base a esta idea, se propuso, probó y desarrolló una técnica de caracterización de audio que logra resultados muy satisfactorios en diferentes tareas de identificación de audio. El problema de caracterización está dividido en dos partes, el proceso de extracción de características y el proceso de representación de los datos. Básicamente, la mayoría del tiempo se dedicó al estudio y desarrollo del proceso de extracción de características, ya que como se mencionó en este trabajo, de éste depende en gran parte la confiabilidad de un sistema de identificación de audio. El proceso de extracción de características tuvo como conceptos principales la *entropía* de Shannon, los *valores de croma* y la descomposición en valores propios de una matriz, que si bien no son conceptos nuevos, si lograron resolver de una mejor manera las necesidades que tiene un sistema de identificación de audio. En concreto, se lograron aportaciones en cuanto a la robustez y tasas de reconocimiento tanto en línea como fuera de línea, que puede llegar a tener un sistema de identificación de audio.

Por otra parte, dos de las aportaciones al estado del arte están relacionadas con el problema de reconocimiento de audio. El problema de reconocimiento está ligado

a la aplicación o uso final que tiene el sistema para la identificación de audio. Por ejemplo, en sistemas de identificación de voz el tema de alineamiento de señales es importante para el proceso de reconocimiento. Para este tema se diseñó una herramienta que da una solución alternativa a este problema. Esta herramienta es una técnica de alineamiento que se basó en la distancia Coseno y sirvió para medir la similitud entre dos series de tiempo. La idea de generar esta herramienta nació de un problema particular del área de la ingeniería biomédica que tiene que ver con el alineamiento de señales fisiológicas. La motivación fue alinear series de tiempo con un algoritmo simple de implementar y de rápida evaluación. En base a esto se empezó la experimentación y se logró generar esta herramienta que tiene gran versatilidad. La principal contribución aquí fue lograr alinear secuencias de series de tiempo con un algoritmo que tiene un tiempo de complejidad lineal. Esta característica es muy importante ya que está ligada con el tiempo de búsqueda del sistema.

Finalmente, la otra aportación al problema de reconocimiento trató el tópico del alineamiento de audio en tiempo real. Por lo general, este tipo de tópicos requieren estar acompañados de diferentes fundamentos para lograr su objetivo. En este problema se consideró un sistema de seguimiento de audio para probar las ideas conceptuales que planteamos como hipótesis. Los experimentos consistieron en identificar el número de estados estimados correctamente usando los PMPO y HMM. Para esto se diseñó un modelo estocástico que interpreta las observaciones del sistema para estimar el estado actual del sistema. El diseño del modelo requirió de tiempo y esfuerzo, ya que no fue fácil adaptar un concepto relacionado con la inteligencia artificial con el modelado de señales de audio. Este trabajo lo consideramos una de las aportaciones más importantes tanto para nosotros como para el área de los últimos años. Como se mencionó dentro de este documento, este tópico es poco referenciado en la literatura por la complejidad que presenta al utilizar la perspectiva manejada aquí, sin embargo, aunque es cierto que faltan algunos experimentos por realizar, estamos convencidos que el camino que llevamos es el correcto.

6.2. Logros

En el Capítulo 3 se presentó nuestra aportación sobre caracterización robusta de señales de audio. En particular se introdujo una nueva técnica de extracción de características la cual se llamó "*entropía por croma*". Los productos generados a partir de esta aportación fueron dos publicaciones en conferencias internacionales especializadas en el área de procesamiento de señales. Una de las conferencias se llevó a cabo en Bilbao,

España dentro del marco del IEEE International Symposium on Signal Processing and InformationTechnology en 2011. La otra conferencia se llevó a cabo en Berlín, Alemania dentro del marco de la 2014 37th International Conference on Telecommunications and Signal Processing. También, una publicación en revista sobre esta aportación está por definirse en los próximos meses.

En el Capítulo 4 se presentó nuestra aportación sobre técnicas de alineamiento. En particular se introdujo una nueva técnica de alineamiento la cual se llamó "*Técnica de Alineamiento por Distancia Coseno*". Los resultados generados con esta aportación están reflejados en dos publicaciones, una en conferencia internacional y otra en revista. La conferencia se llevó a cabo en Morelia, México dentro del marco del 17th International Congress on Computer Science Research en 2011. La revista que considera esta aportación es la International Journal of Combinatorial Optimization Problems and Informatics, Vol. 4, No. 1.

En el Capítulo 5 se presentó nuestra aportación sobre alineamiento de audio en tiempo real. En particular se introdujeron dos tópicos, uno relacionado con el proceso de caracterización de la señal de audio y el otro al modelo para hacer alineamiento de audio. Los resultados de esta contribución se encuentran publicados dentro del IEEE International Autumn Meeting on Power, Electronics and Computing en 2013. Una nueva publicación sobre este tema está por definirse en los próximos meses.

6.3. Trabajo Futuro

En el Capítulo 5 se introdujo el análisis de descomposición de valores propios el cual sirve para caracterizar la señal de audio de manera robusta usando las componentes del vector asociado al valor propio dominante. Sin embargo, sería interesante explorar y comparar este análisis con otros semejantes como son el análisis de componentes principales, el análisis de valores singulares y el análisis de componentes independientes. Nosotros estamos convencidos que estos análisis reportarán conocimiento nuevo a este tópico.

En el Capítulo 5 también se introduce un método para hacer seguimiento de audio mediante un *proceso de Markov parcialmente observable*. Este proceso fue probado usando un modelo discreto sobre el conjunto de observaciones. Como trabajo futuro se prevé revisar este proceso usando un modelo continuo sobre este conjunto. Finalmente, se revisará el desempeño de estos sistemas incrementando la dimensionalidad del modelo mediante la incursión de todas las componentes del vector propio dominante.

REFERENCIAS

[Allamanche01] Allamanche, E., Herre, J., Hellmuth, O., Froba, B., Kastner, T., y Cremer, M. Content based identification of audio material using mpeg-7 low level description. *Proceedings of the International Conference Music Information Retrieval*, 2001.

[Anderberg73] Anderberg, M. *Cluster Analysis for Applications*. Academic Press, New York, 1973.

[Aucouturier07] Aucouturier, J., Defreville, B., y Pachet, F. The bag-of-frame approach to audio pattern recognition: A sufficient model for urban soundscapes but not only for polyphonic music. *Journal of Acoustical Society of America*, 2007.

[Bailey90] Bailey, J. J., Berson, A. S., y Garson, A. Recommendations for standardization and specifications in automated electrocardiography: bandwidth and digital signal processing. *A report for health professionals by an ad hoc writing group of the Committee on Electrocardiography and Cardiac Electrophysiology of the Council on Clinical Cardiology, American Heart Association*, 81, 1990.

[Bello05] Bello, J. P. y Pickens, J. A robust mid-level representation for harmonic content in music signals. *International Symposium on Music Information Retrieval*, págs. 304–311, 2005.

[Bilmes98] Bilmes, J. A. A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *Technical Report, TR-97-021,International Computer Science Institute*, 1998.

[Brown92] Brown, J. C. y Puckette, M. S. An efficient algorithm for the calculation of a constant q transform. *Journal of the Acoustical Society of America*, 92, no. 5:2698–2701, 1992.

[Camarena06] Camarena, J. A. y Chávez, E. On musical performances identification, entropy and string matching. *Lecture Notes in Computer Science*, 4293:952–962, 2006.

[Camarena09] Camarena, J. A. y Chavéz, E. Robust audio-fingerprinting with spectral entropy signatures. *Proceedings of CIARP 14th*, 2009.

[Camarena10] Camarena, J. A. y Chávez, E. Real time tracking of musical performances. *Lecture Notes in Computer Science*, 6438:138–148, 2010.

[Cano99] Cano, P., Loscos, A., y Bonada, J. Score-performance matching using hmms. *Proceedings of the International Computer Music Conference*, págs. 441–444, 1999.

[Cano05a] Cano, P., Batlle, E., Gómez, E., Gómes, L., y Bonnet, M. Audio fingerprinting: Concepts and applications. *International Conference on Fuzzy Systems Knowledge Discovery*, págs. 233–245, 2005.

[Cano05b] Cano, P., Batlle, E., Kalker, T., y Haitsma, J. A review of audio fingerprinting. *The Journal of VLSI Signal Processing*, 41:271–284, 2005.

[Cassandra94] Cassandra, A. R., Kaelbling, L. P., y Littman, M. L. Acting optimally in partially observable stochastic domains. *Proceeding of the Twelfth National Conference on Artificial Intelligence*, págs. 1023–1028, 1994.

[Chen10] Chen, Y. y Gupta, M. Em demystified: An expectation maximization tutorial. *UWEE Technical Report*, Number UWEETR-2010-0002, 2010.

[Dannenberg03] Dannenberg, R. B. y Hu, N. Polyphonic audio matching for score following and intelligent audio editors. *Proceedings of the 2003 International Computer Music Conference*, págs. 27–33, 2003.

[Dixon05] Dixon, S. Live tracking of musical performances using on-line time warping. *Proceedings of the 8th International Conference on Digital Audio Efects*, 2005.

[Duda01] Duda, R. O., Hart, P. E., y Stork, D. G. *Pattern Classification*. Wiley-Intersciance, New York, 2001.

[Everrit93] Everrit, B. *Cluster Analysis*. Halsted Press, New York, 1993.

[Fitzgerald06] Fitzgerald, D., Cranitch, M., y Cychowski, M. T. Towards and inverse constant q transform. *120th Audio Engineering Society Convention*, 2006.

[Fuente10] Fuente, J. L. *Técnicas de Cálculo para Sistemas de Ecuaciones, Programación Lineal y Programación Entera*. Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros Industriales, 2010.

[Gevins97] Gevins, A. High resolution eeg mapping of cortical activation related to working memory: effects of task difficulty, type of processing and practice. *Cereb. Cortex*, págs. 374–385, 1997.

[Graziosi04] Graziosi, D. B., Santos, C. N. D., Netto, S. L., y Biscainho, L. W. A constant q spectral transformation with improved frequency response. *Proceedings of the IEEE International Symposium on Circuits and Systems*, 5:544–547, 2004.

[Gulmezoglu99] Gulmezoglu, M. B., Dzhafarov, V., Keskin, M., y Barkana, A. A novel approach to isolated word recognition. *IEEE Transactions on Speech and Audio Processing*, 1999.

[Gunopulos00] Gunopulos, D. y Das, G. Time series similarity measures. *In tutorial notes of the sixth ACM SIGKDD international conference on knowledge discovery and data mining*, págs. 243–307, 2000.

[Guojun07] Guojun, G., Chaoqun, M., y Jianhong, W. *Data Clustering: Theory, Algorithms and Applications*. Siam, Philadelphia, 2007.

[Haitsma02] Haitsma, J. y Kalker, A. A highly robust audio fingerprinting system. *Proceedings of the International Symposium on Music Information Retrieval*, 2002.

[Hanley82] Hanley, J. A. y McNeil, B. J. The meaning and use of the area under a receiver operating characteristic curve. *Radiology*, págs. 29–36, 1982.

[Hanna08] Hanna, P., Robine, M., Ferraro, P., y Allali, J. Improvements of alignment algorithms for polyphonic music retrieval. *Proceedings of the International Computer Music Modeling and Retrieval Conference*, págs. 244–251, 2008.

[Ibe08] Ibe, O. *Markov Processes for Stochastic Modeling*. Academic Press, United States, 2008.

[Itakura75] Itakura, F. Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, págs. 67–72, 1975.

[Kaplow10] Kaplow, R. *Point Based POMDP solvers: Survey and Comparative Analysis*. Master's Thesis, McGill University, 2010.

[Kotsifakos12] Kotsifakos, A., Papapetrou, P., Hollmén, J., Gunopulos, D., y Athitsos, V. A survey of query by humming similarity methods. *Conference on Pervasive Technologies Related to Assistive Environment*, 2012.

[Logan00] Logan, B. Mel frequency cepstral coefficients for music modeling. *IEEE Proceedings of the Symposium on Music Information Retrieval*, 2000.

[Misra04] Misra, H., Ikbal, S., Bourland, H., y Hermansky, H. Spectral entropy based feature for robust asr. *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, págs. 193–196, 2004.

[Misra05] Misra, H., Ikbal, S., Sivadas, S., y Bourland, H. Multi-resolution spectral entropy feature for robust asr. *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, págs. 253–256, 2005.

[Mohammad94] Mohammad, A. Entropy in signal processing. *Traitement du signal*, págs. 87–116, 1994.

[Mottio08] Mottio, R. y Orio, N. A music identification system based on chroma indexing and statistical modeling. *Proceedings of the International Conference of Music Information Retrieval*, págs. 301–306, 2008.

[Muller07] Muller, M. Dynamic time warping. *Information Retrieval for Music and Motion*, 2007.

[Orio03] Orio, N., Lemouton, S., y Schwarz, D. Score following: State of the art and new developments. *Proceedings of the Conference of New Interfaces for Musical Expression*, págs. 36–41, 2003.

[Pickens01] Pickens, J. A survey of feature selection techniques for music information retrieval. *Technical Report, Center for Intelligent Information Retrieval, Departament of Computer Science, University of Massachussetts*, 2001.

[Rabiner78] Rabiner, L. R. y Schafer, R. W. Digital processing of speech signals. *Prentice Hall*, 1978.

[Rabiner89] Rabiner, L. R. A tutorial on hidden markov models and selected applications in speech recognition. *IEEE Proceedings*, 77:257–286, 1989.

[Rényi61] Rényi, A. On measures of entropy and information. *Proceedings of the Fourth Berkeley Symposium on Mathematics, Statistic and Probability*, 1:547–561, 1961.

[Russell10] Russell, S. J. y Norvig, P. *Artificial Intelligence: A Modern Approach*. Prentice Hall, New York, 2010.

[Sadit12] Sadit, E. y Chavéz, E. Practical proximity searching in large metric databases, 2012. Thesis from Universidad Michoacana de San Nicolás de Hidalgo.

[Sakoe78] Sakoe, H. y Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustic, Speech and Signal Processing*, ASSP-26, no.1:43–49, 1978.

[Salcedo07] Salcedo, D. F. *Modelos Ocultos de Markov: Del reconocimiento de voz a la música*. LULU, Madrid, 2007.

[Salton83] Salton, G. y McGill, M. *Introduction to Modern Information Retrieval*. McGraw Hill, New York, Tokyo, 1983.

[Shannon48] Shannon, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948.

[Shepard64] Shepard, R. N. Circularity in judgements of relative pitch. *The Journal of the Acoustical Society of America*, 36:2346–2353, 1964.

[Smith05] Smith, T. y Simmons, R. Point based pomdp algorithms:improved analysis and implementation. *Proceedings Uncertainty in Artificial intellegence*, 2005.

[Stein09] Stein, M., Schubert, B. M., Gruhne, M., Gatzsche, G., y Mehnert, M. Evaluation and comparison of audio chroma feature extraction methods. *Proceedings of the Audio Engineering Society*, 2009.

[Trebbe95] Trebbe, H. Errata in rabiner's hmm tutorial. *Technical Report, University of Munster*, 1995.

[Typke05] Typke, R., Wiering, F., y Veltkamp, R. A survey of music information retrieval systems. *Proceedings of the International Conference Music Information Retrieval*, págs. 153–160, 2005.

[Venkatachalam04] Venkatachalam, V., Cazzanti, L., Dhillon, N., y Wells, M. Automatic identification of sound recordings. *IEEE Signal Processing Magazine*, 21, no.2:92–99, 2004.

[Wang03] Wang, A. An industrial strength audio search algorithm. *Proceedings of the International Conference on Music Information Retrieval*, págs. 713–718, 2003.

[Wold96] Wold, E., Blum, T., Keislar, D., y J.Wheaton. Content-based classification, search and retrieval of audio. *IEEE Multimedia*, 3:27–36, 1996.

[Yu08] Yu, H. M. y Tsai, W. H. A query by singing system for retrieving karaoke music. *IEEE Transactions on multimedia*, 10:1626–1637, 2008.

[Özer05] Özer, H., Sankur, B., Memon, N., y Anar, E. Perceptual audio hashing functions. *EURASIP Journal on Applied Signal Processing*, 12:1780–1793, 2005.

[Zhang03] Zhang, B. y Srihari, S. Properties of binary vector dissimilarity measures, 2003. Technical Report, CEDAR Department of Computer Science and Engineering, University of Buffalo.

[Zhu06] Zhu, Y. y Kankanhalli, M. S. Precise pitch profile feature extraction from musical audio for key detection. *IEEE Transactions on Multimedia*, 8, no. 3:575–584, 2006.