UNIVERSIDAD NACIONAL DEL LITORAL
Facultad de Ingeniería y Ciencias Hídricas

Optimización mediante algoritmos evolutivos de la representación de señales para el reconocimiento automático del habla

Leandro Daniel Vignolo

Tesis remitida al Comité Académico del Doctorado
como parte de los requisitos para la obtención
del grado de
DOCTOR EN INGENIERIA
Mención Inteligencia Computacional, Señales y Sistemas
de la
UNIVERSIDAD NACIONAL DEL LITORAL

2011

Comisión de Posgrado, Facultad de Ingeniería y Ciencias Hídricas, Ciudad Universitaria,
Paraje "El Pozo", S3000, Santa Fe, Argentina.

Doctorado en Ingeniería
Mención Inteligencia Computacional, Señales y Sistemas

Título de la obra:

Optimización mediante algoritmos evolutivos de la representación de señales para el reconocimiento automático del habla

Autor: Leandro Daniel Vignolo
Director: Dr. Hugo Leonardo Rufiner
Codirector: Dr. Diego Humberto Milone

Lugar: Santa Fe, Argentina

Palabras Claves:

algoritmos evolutivos,
cuantización vectorial,
modelos ocultos de Markov,
paquete de onditas,
coeficientes cepstrales,
reconocimiento robusto del habla.

Dedicado a Patricia, mi otra mitad.

Dedicado a mis padres Angela y Miguel.

Dedicado a mis sobrinos Valentino, Giuliano y Delfina.

Agradecimientos

Quiero expresar mi agradecimiento a mi director, Dr. Leonardo Rufiner, por compartir conmigo su experiencia y permitirme enriquecer esta tesis con sus puntos de vista. En igual medida quiero agradecer a mi co-director Dr. Diego Milone, por darme la posibilidad de conocer el mundo de la investigación, por guiarme a lo largo de todo el trabajo con suma dedicación, por sus valiosos aportes y su apoyo constante. A ambos, Diego y Leonardo, por brindarme su confianza y la oportunidad de trabajar en el **sinc**(i), por las numerosas discusiones sobre la tesis, y por contribuir en mi formación de manera ejemplar.

También quiero agradecer al Dr. John Goddard, por haber aportado ideas muy valiosas para el desarrollo de la tesis y los trabajos relacionados con ésta, por hacer posible mi estancia de investigación en la UAM, y por su hospitalidad y cordialidad durante mi estadía.

Quiero agradecer también a todos mis compañeros del **sinc**(i) por crear un ambiente de trabajo ideal. Particularmente a Leandro Di Persia, por su compañerismo y predisposición, por brindarme siempre su asistencia, facilitando mi aprendizaje y allanando mi camino. A Marcelo, por acompañarme en esta carrera y estar siempre dispuesto a brindarme su ayuda. También les quiero agradecer a César, Leonardo, Diego, Carlos, Matías, Federico, Cecilia y Maximiliano, por su compañerismo y amistad. A la Dra. María Eugenia Torres, por estar siempre dispuesta a ayudar con sus consejos y sugerencias.

Les agradezco también a mis padres, Angela y Miguel, por su ejemplo, su comprensión, y por toda la ayuda que me han brindado siempre, haciendo posible mi dedicación a esta carrera. También quiero agradecer de manera especial a Patricia, por completarme, por estar a mi lado en los momentos difíciles, por su comprensión, por darme su apoyo, aliento y fuerzas para seguir adelante. A mis hermanos, María de los Ángeles y Damián, quienes siempre me han brindado su ayuda incondicional, y a mis sobrinos Delfina, Giuliano y Valentino por alegrarme con su ternura. A mi abuela Ana y mi tío Héctor por su cariño y sus oraciones, y a todos mis amigos por su afecto.

También deseo expresar mi agradecimiento a las siguientes instituciones:

- **sinc**(i): Centro de Investigación en Señales, Sistemas e Inteligencia Computacional
- Facultad de Ingeniería y Ciencias Hídricas, Universidad Nacional del Litoral

- Consejo Nacional de Investigaciones Científicas y Técnicas
- Laboratorio de Cibernética (Facultad de Ingeniería, Universidad Nacional de Entre Ríos)
- Departamento de Ingeniería Eléctrica, Unidad Iztapalapa, Universidad Autónoma Metropolitana, México

A todos, mi más sincero agradecimiento,

Leandro Daniel Vignolo
Santa Fe, Abril de 2011

Optimización mediante algoritmos evolutivos de la representación de señales para el reconocimiento automático del habla

Leandro Daniel Vignolo

Director de tesis: Hugo Leonardo Rufiner
Co-director de tesis: Diego Humberto Milone
Departamento de Informática, 2011

Resumen

La dificultad para resolver los problemas asociados al reconocimiento del habla está dada por las características de las señales implicadas, ya que las mismas presentan complejas funciones de densidad de probabilidad, son no estacionarias y generalmente se encuentran contaminadas con ruidos de naturaleza e intensidad muy diversa. Es por esto que los sistemas de reconocimiento automático requieren de una etapa de procesamiento que ponga en evidencia las características distintivas de cada fonema, permitiendo mejorar los resultados. Se considera que el hecho de lograr una representación robusta ante el ruido es de fundamental importancia para facilitar la tarea de reconocimiento. Existen diferentes técnicas de procesamiento como el análisis espectral por bandas, los métodos de predicción lineal y el análisis cepstral. Sin embargo, estas técnicas han fracasado en general para el caso de señales con ruido y son de escasa utilidad fuera de las condiciones de laboratorio.

El objetivo de esta tesis es el desarrollo de un método para optimizar la etapa de procesamiento de la señal de voz, de manera que permita mejorar los resultados de un sistema de reconocimiento automático del habla. Dicho método consiste en la aplicación de algoritmos evolutivos para optimizar el vector de características utilizado para representar las señales de voz. Se parte de la hipótesis de que cuanto mejor sea el análisis o proceso utilizado para generar los patrones a identificar, más separadas quedarán las clases en el espacio de características y la tarea de clasificación resultará más sencilla.

En los últimos años se han realizado nuevos avances en el desarrollo de técnicas de extracción de características robustas al ruido. Muchos de estos se han basado en la introducción de modificaciones en la forma de calcular los coeficientes cepstrales en escala de mel. En esta tesis se propone continuar la búsqueda de una representación basada en coeficientes cepstrales, mediante la optimización del banco de filtros utilizado para el cálculo de los mismos. Dicha búsqueda es dirigida por el resultado de clasificación de fonemas, obtenido para cada posible solución.

Por otro lado, recientemente han cobrado importancia las representaciones de señales basadas en onditas. Esto se debe a que sus características las hacen útiles para

el análisis de señales no estacionarias. Con el procesamiento basado en la descomposición completa de la transformada paquete de onditas se obtiene un conjunto redundante de coeficientes que representan una señal. Si bien estos coeficientes resultan útiles para la clasificación, la gran cantidad de los mismos dificulta la tarea del clasificador. A este problema, que surge cuando se trata con espacios de características de grandes dimensiones, se lo conoce como *maldición de la dimensionalidad*: la cantidad de datos necesarios para el entrenamiento crece exponencialmente con la cantidad de dimensiones. Es por esto que resulta conveniente encontrar un subconjunto de coeficientes que permita, de la mejor manera posible, identificar las distintas clases que éstas presentan. Para encontrar una representación más apropiada para la clasificación de fonemas se propone, al igual que en el caso anterior, la utilización de un algoritmo evolutivo.

Los resultados obtenidos en la clasificación de distintos conjuntos de fonemas muestran que ambos métodos desarrollados cumplen el objetivo de encontrar una representación que permite mejorar el desempeño respecto a las parametrizaciones tradicionales. Además, en las pruebas realizadas se consideraron señales con distintas cantidades de ruido, permitiendo comprobar la robustez de las representaciones optimizadas.

Evolutionary optimization of signal representations for automatic speech recognition

Leandro Daniel Vignolo

Thesis director: Hugo Leonardo Rufiner
Thesis co-director: Diego Humberto Milone
Departamento de Informática, 2011

Abstract

The key issue on speech recognition is given by the characteristics of the signals involved, as these are governed by complex probability density functions, are non-stationary and generally contaminated with noise of diverse nature and intensity. This is why the automatic recognition systems need a processing stage in order to bring out the key features of phonemes, allowing to improve their performance. It is considered that the robustness of the representation to environmental noise is of key importance in the recognition task. There are different processing techniques such as spectral band analysis, linear prediction and cepstrum. However, these techniques have mostly failed in the case of noisy signals and, therefore, are not useful outside the laboratory.

The goal of this thesis is the development of a methodology for the optimization of the signal processing stage, in order to improve the results of an automatic speech recognition system. This methodology consists in the use of evolutionary algorithms for the optimization of the feature vector used for speech signal representation. The hypothesis is that the better the analysis or process applied to the patterns that are to be classified, the more separated would the classes result in the features space and, therefore, the classification task would be simpler.

During the last years, several advances had been made in the development of noise robust feature extraction techniques. Many of these were based on the mel scaled cepstral coefficients, and introduced modifications to this feature extraction technique. In this thesis, the first proposal is to continue the search for an optimal representation based on cepstral coefficients, by the optimization of the filter-bank involved in this feature extraction procedure. This search is conducted by the phoneme classification result, which is obtained for each candidate solution.

On the other hand, wavelets have recently gained importance in the signal processing fields. This is because they have characteristics that are useful for the analysis of non-stationary signals. By means of the full decomposition provided by the wavelet packets transform a redundant set of coefficients is obtained. These features present discriminative information, however, the large number of coefficients makes the task of the classifier more difficult. This problem, which arises when dealing with high dimensional

feature spaces, is known as *curse of dimensionality*: the amount of data needed for training grows exponentially with the number of dimensions. Because of this, it is convenient to find the subset of coefficients which maximizes the discrimination capability. In order to find a speech representation more appropriate for phoneme classification, as in the first proposal, the use of an evolutionary algorithm is proposed.

The results obtained in the classification of different phoneme sets show that both of the approaches proposed in this thesis meet the objective of finding a good representation to improve the performance of the classical speech features. Moreover, in order to evaluate the robustness of the optimized representations, they were evaluated in the phoneme classification task for different noise conditions.

Índice general

Índice de tablas

Índice de figuras

Prefacio

Desde hace alrededor de cinco décadas se pretende lograr la comunicación oral entre personas y máquinas. Alcanzar dicho objetivo supone que las computadoras sean capaces de decodificar e interpretar un mensaje a partir de las ondas de presión sonoras que lo transportan.

En un principio se pensó que este proceso, que resulta natural y sencillo para las personas, podría ser fácilmente imitado mediante programas de computadoras. Sin embargo, luego de haberse logrado cierto éxito en el reconocimiento automático de palabras aisladas, se abordó con optimismo el problema de reconocimiento en discurso continuo para descubrir la enorme complejidad del problema. Desde entonces se ha invertido en importantes y ambiciosos proyectos de investigación, surgiendo así una actividad multidisciplinaria en la que intervienen ingenieros informáticos y electrónicos, lingüistas y fonetistas, entre otros. A pesar de ésto, los sistemas actuales de reconocimiento automático aún tienen importantes restricciones para poder brindar resultados aceptables a los fines prácticos. Entre ellas se pueden citar: la pronunciación de palabras en forma aislada, vocabularios limitados, dependencia del locutor, escasez de ruido ambiental, etc.

Los problemas que hacen difícil esta tarea vienen dados por las propias características del habla, como la continuidad y la variabilidad en la pronunciación, así como también por las condiciones ambientales que suelen afectar la percepción de la señal de voz. A pesar de los importantes avances logrados, el desempeño de los sistemas artificiales se ve muy afectado por el ruido de fondo, lo que dificulta en gran medida su utilización en ambientes no ideales. Para obtener sistemas más robustos, es decir, que su desempeño no sea tan sensible a las condiciones reales de ruido, gran parte de los esfuerzos actuales se centran en mejorar la etapa de procesamiento de señales. Más precisamente, en desarrollar técnicas de procesamiento que capturen la información relevante, pero al mismo tiempo minimicen los efectos del ruido en el conjunto de características resultante.

En esta tesis se plantea la búsqueda de una representación más apropiada para la clasificación de fonemas como un problema de optimización. Se propone una estrategia evolutiva que permite adaptar una representación, mediante la optimización de un conjunto de parámetros, y valiéndose de la información implícita en los datos. Con este nuevo enfoque se puede obtener un conjunto alternativo de características que provea mayor robustez y permita mejorar los resultados de un sistema de reconocimiento automático del habla.

Los objetivos perseguidos en esta tesis son los que se detallan a continuación:

- Diseñar una metodología para *optimizar* la etapa de *extracción de características* de un sistema de reconocimiento automático del habla.
- Optimizar una representación para las señales de voz partiendo de técnicas clásicas de procesamiento.
- Explorar la optimización de representaciones que no fueron originalmente diseñadas para el habla.
- Encontrar una *representación* de señales de habla que permita mejorar los resultados de clasificación obtenidos con las técnicas tradicionales de procesamiento para un conjunto de fonemas.
- Comprobar la robustez de las representaciones obtenidas mediante pruebas de validación y señales con diferentes cantidades de ruido.
- Incorporar dicha representación en un sistema de reconocimiento del habla para mejorar el desempeño en condiciones adversas.

La organización del documento de la tesis es la siguiente. En el Capítulo 1 se presenta una breve introducción al reconocimiento automático del habla. Luego se ofrece una revisión de los métodos de aprendizaje maquinal empleados en este trabajo y una descripción de los algoritmos evolutivos.

En el Capítulo 2 se presentan las nociones básicas sobre la señal de voz y el procesamiento de la misma, y se desarrolla una revisión de las técnicas de análisis más utilizadas, como marco conceptual de las propuestas del trabajo.

En el Capítulo 3 se desarrolla el primer aporte de la tesis, que propone la utilización de algoritmos evolutivos para encontrar una representación cepstral robusta. Esta propuesta de optimización se divide a su vez en dos estrategias diferentes. En primer lugar se plantea la codificación directa de los parámetros del banco de filtros empleado en el cálculo de los coeficientes cepstrales. Luego se propone una alternativa de codificación para reducir la cantidad de parámetros, que facilita la convergencia del algoritmo evolutivo.

Una segunda estrategia evolutiva se propone en el Capítulo 4. En este caso, la optimización se plantea a partir de la descomposición completa de la transformada paquete de onditas, para encontrar el conjunto de coeficientes que mejor represente a la señal de voz. Se discuten los resultados obtenidos y se comparan con los resultados de la estrategia del capítulo anterior.

Finalmente, en el Capítulo 5 se presentan las conclusiones particulares y generales de las dos propuestas de la tesis, y se enumeran algunos de los trabajos que se considerarán en el futuro. Además, en el Apéndice A se describe una estrategia paralelización del algoritmo evolutivo, la cual permitió reducir drásticamente el tiempo necesario para la experimentación numérica relacionada con la tesis.

Capítulo 1

Introducción

1.1. Motivación

Desde hace tiempo se estudia la posibilidad de desarrollar interfaces hombre-máquina controladas por voz para sustituir, en ciertas aplicaciones, a las interfaces tradicionales basadas en teclados, paneles o dispositivos similares. La utilización de la voz como vía para interactuar con las computadoras ofrece varias ventajas respecto al método tradicional de comunicación entre el usuario y la máquina. El objetivo del reconocimiento automático del habla (RAH) es hacer posible la comunicación hablada entre seres humanos y computadoras.

En el estudio del habla intervienen diversas áreas de conocimiento (acústica, fonética, fonológica, semántica, etc.), cada una de las cuales juega un rol importante en los sistemas de reconocimiento automático. La fisiología, por su parte, mediante el estudio de los órganos que participan en la producción y la percepción de la voz, ha permitido comprender las características particulares de esta señal y proponer distintas representaciones que lograron favorecer el reconocimiento. Luego, distintos campos de las ciencias de la computación, como el procesamiento de señales y la inteligencia artificial brindan las herramientas que permiten hacer cooperar estas distintas fuentes de conocimiento en un sistema de RAH.

Desde los comienzos del RAH se ha observado que utilizando la señal acústica de la voz directamente no se explota toda la información que hay implícita en la misma. Por esta razón, se han ido incorporando distintos niveles de análisis del habla, haciendo explícitas cada vez más características de esta señal. De esta manera, progresivamente, se ha logrado mejorar el rendimiento de los sistemas de RAH, incorporando información fonética, fonológica, prosódica, gramatical, etc.

Actualmente, en la mayoría de los sistemas de RAH las señales de voz se procesan mediante una parametrización ya tradicional, basada en los coeficientes

cepstrales en escala de mel [Davis y Mermelstein, 1980]. Esta representación biológicamente inspirada se basa en el uso de una escala psicoacústica para imitar la respuesta en frecuencia del oído humano. Sin embargo, como el sistema auditivo humano es complejo y aún no se ha comprendido completamente su funcionamiento, los parámetros para obtener una representación óptima son desconocidos. Por otro lado, el desempeño de los reconocedores actuales se ve degradado rápidamente a medida que aumenta la cantidad de ruido en la señal. Esto ha motivado el desarrollo de nuevas alternativas para la parametrización de las señales de habla, algunas de las cuales se obtienen mediante procesos similares al de la representación clásica. Otras se basan en técnicas de procesamiento de señales que son relativamente nuevas en el área de reconocimiento del habla.

Una estrategia habitual es la de optimizar experimentalmente una representación a partir de un conjunto de datos de habla. Esto se realiza usualmente ajustando parámetros de algún modelo de manera de mejorar el desempeño del sistema de reconocimiento [Hermansky, 1998]. Es decir, el conocimiento a priori acerca de las características de la señal de voz, subyacente en el modelo, se combina con la información obtenida experimentalmente a partir de los datos. Siguiendo dicho enfoque, en esta tesis se introduce una metodología basada en algoritmos evolutivos para constituir un aporte en la búsqueda de una mejor representación para las señales de voz. La optimización es guiada según la minimización del error en la clasificación de un conjunto de fonemas, que es una tarea relacionada directamente al reconocimiento.

Este capítulo se organiza de la siguiente manera. En la Sección 1.2 se introducen algunas nociones básicas sobre el RAH y las etapas que lo componen. En la Sección 1.3 se describen los dos algoritmos de clasificación empleados en la tesis, uno basado en la cuantización vectorial y otro basado en los modelos ocultos de Markov. A continuación, en la Sección 1.4 se presentan los algoritmos evolutivos y se enumeran las características que los hacen apropiados para su aplicación en el procesamiento de señales.

1.2. Reconocimiento automático del habla

1.2.1. El lenguaje y el habla

Una de las manifestaciones más importantes de la inteligencia del ser humano es la comunicación mediante el lenguaje natural, que le permite expresar ideas, comunicarse, establecer relaciones, etc. Si bien el hombre es capaz de llevar a cabo la comunicación sin dificultades, la comprensión del lenguaje es una tarea de gran complejidad. De hecho, para comprender un mensaje, el ser humano considera información adicional a la onda sonora que oye. Ésta es una de las razones por las que una persona puede entender un mensaje sin inconvenientes aún cuando la señal de voz está perturbada con ruido.

Gracias al enorme desarrollo de las tecnologías y las ciencias de la computación, en las últimas décadas el estudio del RAH ha experimentado importantes avances, sin embargo, las limitaciones siguen siendo importantes. Es decir, sólo cuando el problema se simplifica considerando palabras aisladas, limitando el vocabulario o la cantidad de hablantes, el reconocimiento puede ser realizado por una computadora con un nivel de desempeño aceptable.

1.2.2. Orígenes del reconocimiento automático del habla

Los primeros sistemas de RAH surgieron en los años 50, luego del desarrollo del conversor analógico-digital de alta velocidad. Estos primeros trabajos abordaban el reconocimiento con un vocabulario reducido, del orden de 10 palabras, emitidas por un único locutor. Posteriores avances resultaron en la inclusión de reglas fonotácticas [Church, 1983] y algoritmos de alineamiento dinámico del tiempo para reconocimiento de patrones explícitos [Sakoe y Chiba, 1978]. A principio de la década del 70, James Baker propuso un método alternativo para el RAH [Baker, 1974]. Esta estrategia puramente estadística se basó en los modelos ocultos de Markov. Estos modelan las unidades de sonido como una secuencia de estados, donde cada estado tiene su propia y única distribución de probabilidad. Este sistema de reconocimiento basado en modelos de Markov representó un cambio paradigmático en relación a las investigaciones realizadas hasta ese momento. A finales de la década del 80, surgieron las primeras investigaciones para resolver el problema del reconocimiento de habla continua (frases más o menos largas sin separación entre palabras) [Davis y Mermelstein, 1980]. El sistema Sphinx, desarrollado en la Universidad de Carnegie Mellon, incorporó varias innovaciones en

el modelado del habla y en los algoritmos utilizados para el reconocimiento [Lee et al., 1990]. Éste era un sistema de habla continua e independiente del hablante que no sólo reconocía habla con alta precisión, sino que era capaz de hacerlo en tiempo real. Este sistema demostró al mundo que el reconocimiento automático de habla continua independiente del hablante era posible y realizable con los recursos computacionales de la época.

Desde entonces, la investigación en el campo del RAH ha logrado importante un desarrollo, gracias a los avances en el procesamiento de señales y el aprendizaje maquinal [Rabiner y Juang, 1993]. Hoy en día se utilizan sistemas de RAH en distintas aplicaciones, que abarcan tareas de reconocimiento desde pequeños conjuntos de palabras hasta el discurso continuo con grandes vocabularios. En la Figura 1.1 se puede apreciar como fue evolucionando la tasa de error en palabras para distintas tareas de RAH a lo largo del tiempo. La dificultad de estas tareas, desarrolladas en DARPA (del inglés *defense advanced research projects agency*), varía según el tipo de habla (palabras aisladas, habla leída, habla espontánea, etc.) y el tamaño del vocabulario (expresado en miles de palabras, K). Como se puede apreciar, la tasa de error se ha ido reduciendo en todos los casos [Rabiner y Schafer, 2011].

1.2.3. El problema del reconocimiento

La comunicación mediante la voz es una tarea que el hombre realiza con aparente facilidad. Así mismo, el aprendizaje del idioma materno se realiza de forma espontánea, con un proceso de aprendizaje basado en la imitación y repetición de sonidos.

A pesar de la sencillez que parece presentar el problema del habla, el estudio de la mismo muestra una enorme complejidad. En el habla aparecen mezclados varios niveles de descripción, que interactúan entre si, no existiendo un modelo apropiado para representar dicho proceso. De esta forma, el RAH presenta una naturaleza interdisciplinaria, siendo necesario aplicar técnicas y conocimientos procedentes de las siguientes áreas [Rabiner y Juang, 1993]:

- Procesamiento de señales
- Física (acústica)
- Reconocimiento de patrones
- Teoría de la información y comunicaciones

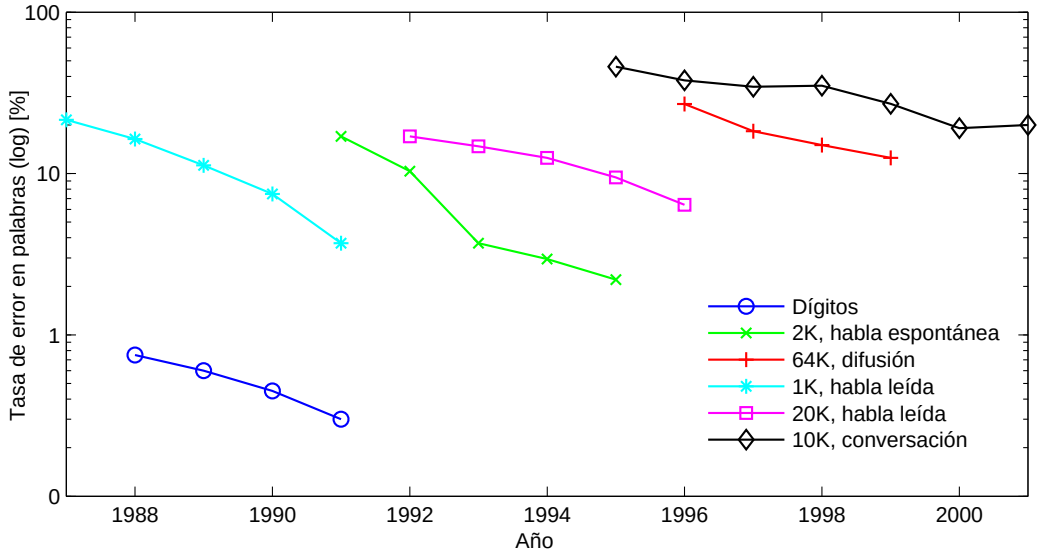


Figura 1.1. Evolución de la tasa de error en palabras para diferentes tareas de RAH a lo largo del tiempo (Fuente [Rabiner y Schafer, 2011]). Para cada tarea se expresa el tamaño del vocabulario en miles de palabras (K).

- Lingüística
- Fisiología
- Informática

En la actualidad existen sistemas comerciales capaces de reconocer voz de forma automática que demuestran la factibilidad de abordar del problema del RAH. A pesar de los avances logrados aún quedan muchos problemas por resolver en el campo del RAH, entre los cuales se puede mencionar [Rabiner y Schafer, 2011]:

- Aumento del tamaño del vocabulario reconocido
- Aumento de la cantidad de hablantes.
- Reconocimiento de habla continua y espontánea.
- Robustez al ruido ambiente.

Los sistemas de RAH actuales se pueden subdividir en dos grupos bien diferenciados por la complejidad del problema que intentan resolver: los que se dedican únicamente al reconocimiento de palabras aisladas (separadas unas de otras por silencios claramente marcados) y los que se dedican al reconocimiento de la habla continua.

En el habla continua, la cantidad de combinaciones de palabras (número de frases) posibles es extremadamente elevada, y no es factible disponer de muestras de todas estas combinaciones para realizar un aprendizaje fiable de cada una. Algo similar ocurre con el aprendizaje de modelos de palabras aisladas cuando el vocabulario crece. Todo ello obliga a que en los sistemas de RAH continua se deba recurrir a la descomposición en sub-unidades (fonemas, palabras, etc.). Esto los lleva inevitablemente a enfrentarse con el problema de la segmentación, que se dificulta por las formas de articulación de las distintas unidades.

1.2.4. La unidad fonética

Debido a la alta variabilidad de las características acústicas de los sonidos según su contexto, es necesario elegir una representación de la señal distinguiendo unidades elementales. Dos tipos de unidades en este sentido son: los alófonos y los fonemas. Los alófonos son las representaciones de los sonidos según aparecen realmente en las palabras. Los fonemas son representaciones más abstractas que capturan las características comunes de una clase de alófonos y que pueden caracterizar sus rasgos acústicos. La mayor ventaja de la utilización de fonemas como unidad de base para representar las palabras habladas es que nunca hay más de 40 fonemas distintos por lengua (del orden de 20 para el castellano) ya que este tipo de representación no tiene en cuenta los rasgos propios del hablante, las emociones u otras fuentes de variabilidad. La desventaja es que en el habla natural la señal acústica real de cada fonema depende del que le precede, del que le sigue y del estado transitorio del aparato fonador. Clasificar a los alófonos en fonemas abstractos requiere un análisis muy fino de la manera en la que el contexto del discurso determina los alófonos de un fonema. Estos están sometidos a alto grado de variación debido a las diferencias entre hablantes, las de un mismo locutor y las producidas por el contexto, además del acento y la entonación.

La dificultad de las técnicas clásicas de procesamiento para resolver los problemas asociados al reconocimiento del habla está dada por la complejidad de las señales implicadas, ya que las mismas presentan funciones estadísticas de densidad superpuestas para las diferentes clases y son no estacionarias.

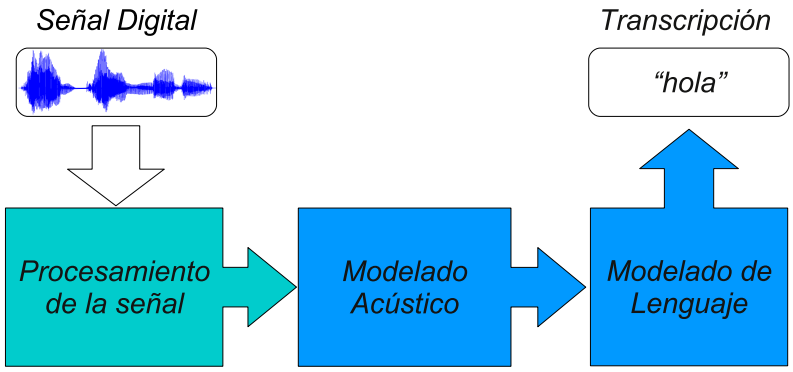


Figura 1.2. Etapas básicas de un sistema de RAH.

1.2.5. Etapas de un sistema de reconocimiento

Como se esquematiza en la Figura 1.2, la estructura general de los sistemas de RAH se compone esencialmente de tres módulos o etapas:

- *Procesamiento y análisis del habla:* en esta etapa se realiza un análisis de la señal de voz en términos de la evolución temporal de parámetros espectrales (previa conversión analógica/digital de la señal). Esto tiene por función hacer más evidentes las características necesarias para la etapa siguiente y a veces también limpiar y reducir la dimensión de los patrones para facilitar su clasificación. En esta etapa, que afecta directamente en los resultados de clasificación en la etapa siguiente así como el resultado final de reconocimiento, se centra el trabajo de esta tesis.

- *Clasificación de unidades fonéticas o modelo acústico:* en esta etapa se clasifica o identifica a los segmentos de voz ya procesados con símbolos fonéticos (fonemas, difonos o sílabas). A veces se puede asociar una probabilidad con este símbolo fonético, lo que permite ampliar la información presentada al siguiente módulo.

- *Análisis en función de reglas del lenguaje o modelo del lenguaje:* en esta última etapa se pueden aprovechar las reglas utilizadas en la codificación del mensaje contenido en la señal para mejorar el desempeño del sistema y producir una transcripción adecuada. Aquí se utilizan otras fuentes de

conocimiento como la ortográfica, la sintáctica, la prosódica, la semántica o la pragmática.

1.3. Métodos de aprendizaje maquinal

Usualmente, con el término aprendizaje maquinal se hace referencia a los cambios que sufre la estructura de un sistema que realiza tareas relacionadas a la inteligencia artificial (reconocimiento, control, predicción, etc.), de tal manera que se pueda esperar un mejor desempeño posterior [Nilsson, 1996]. Este aprendizaje es realizado autónomamente a partir de los ejemplos presentados al sistema, el cual ajusta su estructura interna para generar una salida deseada o bien de capturar las características subyacentes de la distribución de probabilidad de los datos.

La importancia del aprendizaje maquinal se hace evidente, principalmente, en ciertas tareas que sólo pueden definirse a partir de ejemplos. Es decir, aquellos casos en los que no es posible especificar de manera concisa la relación entre las entradas y las salidas deseadas. Más aún, en general y para una determinada tarea, el número de situaciones o ejemplos diferentes suele ser infinito. Por lo tanto, sólo se puede contar con un conjunto de "muestras" que representen en mayor o menor medida la densidad de probabilidad de los datos. Es por esto que el sistema debe ser capaz de generalizar a partir de los ejemplos. Dicha habilidad de obtener la salida correcta para un caso no utilizado en el entrenamiento se denomina capacidad de generalización [Bishop, 2007].

En esta tesis se aplicarán distintos métodos de aprendizaje maquinal en tareas de reconocimiento de patrones, más específicamente en problemas de clasificación. Por lo tanto, a continuación se presentarán, con un enfoque limitado a dicha tarea, dos métodos de aprendizaje maquinal: el primero basado en la cuantización vectorial y el segundo basado en los modelos ocultos de Markov. Vale aclarar que el objetivo aquí no es brindar una revisión completa de dichos métodos, sino el de introducir al lector a las técnicas empleadas en el desarrollo de los capítulos siguientes.

1.3.1. Cuantización vectorial con aprendizaje

Redes neuronales y aprendizaje competitivo

El cerebro humano es capaz de interpretar información imprecisa, suministrada por medio de los sentidos, a una velocidad sorprendente. Es eficaz para realizar acciones perceptuales, como el reconocimiento de rostros y de habla, así como para controlar movimientos corporales. Estas capacidades son desarrolladas explotando las ventajas que brinda el paralelismo masivo de su estructura de procesamiento. Nuestro cerebro está constituido por más de diez billones de neuronas, las cuales se encuentran conectadas entre sí para formar circuitos neuronales. Estos circuitos constituyen sistemas adaptativos que cambian su estructura a partir de la información que reciben como estímulo [Haykin, 1999].

Inspirados en el funcionamiento del cerebro y motivados por la eficiencia con la que el mismo lleva a cabo procesos complejos, desde hace alrededor de 60 años, numerosos investigadores han participado en el desarrollo de las redes neuronales artificiales (ANN, del inglés *artificial neural networks*) [Rutkowski, 2005]. Las ANN son modelos computacionales inspirados en la estructura y los aspectos funcionales de las redes neuronales biológicas [Widrow y Lehr, 1990], y se utilizan para aprender estrategias de solución basadas en ejemplos o patrones de comportamiento.

En este contexto, el proceso de aprendizaje implica la siguiente secuencia de eventos [Haykin, 1994]:

1. La red neuronal es estimulada por el entorno.

2. La red neuronal sufre cambios como resultado a la estimulación.

3. La red neuronal responde de una manera distinta al entorno, como resultado de dichos cambios.

Como las redes neuronales pueden aprender a diferenciar patrones mediante ejemplos y entrenamientos, no es necesario elaborar modelos a priori ni especificar funciones de distribución de probabilidad [Haykin, 1999]. Las redes neuronales son sistemas dinámicos auto-adaptativos, debido a la capacidad de auto-ajuste de los elementos procesales (neuronas) que componen el sistema. Sin embargo, es necesario un buen algoritmo de aprendizaje que, mediante un entrenamiento previo con patrones de las distintas clases, le proporcione a la red la capacidad de discriminar.

El aprendizaje competitivo es una regla en la que, como su nombre lo indica, las neuronas de una red compiten entre ellas para ser activadas. De esta manera, sólo una neurona puede ser activada en cada momento y esta característica la hace apropiada para descubrir los rasgos distintivos que pueden ser utilizados para clasificar un conjunto de patrones. Esto lo hace agrupando los datos que se introducen en la red. De esta forma, las informaciones similares son clasificadas formando parte de la misma categoría, y por tanto deben activar la misma neurona.

Aprendizaje competitivo por cuantización vectorial

La cuantización vectorial es una técnica de procesamiento de señales que permite modelar funciones de densidad de probabilidad mediante la distribución de un conjunto de vectores prototipos. Dado un conjunto de vectores, este es dividido en grupos y luego cada grupo es representado por su centroide [Gray, 1984]. Supongamos que se dispone de un conjunto de vectores o patrones $\mathbf{X} = \{\mathbf{x}(\ell) : \mathbf{x}(\ell) \in \mathbb{R}^n, \ell = 1, 2...\}$ en el conjunto de entrenamiento P, de manera que el sistema aprende secuencialmente y cada paso, t, es determinado por un patrón. En cada paso del aprendizaje, se tiene conjunto fijo de vectores de referencia o prototipos $\{\mathbf{m}_i : \mathbf{m}_i \in P, i = 1, 2 \ldots, N_p\}$, que se modifican según los siguientes principios [Kohonen, 2001]:

1. El conjunto inicial de prototipos $\{\mathbf{m}_i, i = 1, 2, ..., N_p\}$ ha sido inicializado de alguna manera. Por ejemplo, escogiendo prototipos aleatoriamente a partir del conjunto P.

2. En cada paso los patrones $\mathbf{x}(\ell)$ pueden compararse con los prototipos $\mathbf{m}_i$, y el prototipo m_c más cercano a $\mathbf{x}(\ell)$ se actualiza de manera que éste se acerque aún más a $\mathbf{x}(\ell)$.

El prototipo más cercano se encuentra a partir de la siguiente ecuación:

$$\mathbf{m}_c : \delta(\mathbf{x}(\ell), \mathbf{m}_c) = \min_{i=1...N_p} \{\delta(\mathbf{x}(\ell), \mathbf{m}_i)\}, \tag{1.1}$$

donde $\delta(\mathbf{a}, \mathbf{b})$ representa una métrica entre $\mathbf{a}$ y $\mathbf{b}$, como por ejemplo la distancia Euclidea. Se trata entonces de:

a. modificar $\mathbf{m}_c$ de forma que $\delta(\mathbf{x}(\ell), \mathbf{m}_c)$ decrezca,

b. no modificar $\mathbf{m}_i$ cuando $i \neq c$.

Según esta estrategia se modifica un único prototipo en cada paso de aprendizaje (cada patrón modifica un único prototipo). Este procedimiento clásico produce una aproximación de las funciones de densidad de probabilidad de las clases usando un número finito de prototipos. Sea $\mathbf{m}_c$ el prototipo más cercano a $\mathbf{x}(\ell)$ usando una métrica Euclideana, la corrección que se aplica a los prototipos está determinada por:

$$\begin{aligned} \mathbf{m}_c &\leftarrow \mathbf{m}_c + \alpha[\mathbf{x}(\ell) - \mathbf{m}_c], \\ \mathbf{m}_i &\leftarrow \mathbf{m}_i, \quad \text{para } i \neq c \end{aligned} \tag{1.2}$$

donde α, que determina la "cantidad" de corrección, decrece monótonamente durante el entrenamiento y verifica $0 < \alpha < 1$ [Kohonen et al., 1996]. Puede demostrase que este procedimiento de actualización iterativa por la técnica de gradiente descendiente define los valores óptimos de los prototipos asintóticamente [Kohonen, 2001]. No obstante, no es ésta la única estrategia ya que existen alternativas en las que un patrón modifica a más de un prototipo.

Aprendizaje supervisado

La cuantización vectorial con aprendizaje (LVQ, del inglés *learning vector quantization*) supone una extensión del aprendizaje competitivo donde los prototipos están etiquetados [Kohonen, 1990]. Ahora, además de considerar la cercanía de un prototipo se puede evaluar la clase de éste e imponer, por lo tanto, correcciones de premio (acercamiento) o castigo (alejamiento). El objetivo final puede resumirse como sigue: dado P, el conjunto de entrenamiento original, se trata de construir un conjunto de referencia, P_{LVQ}, mediante la técnica LVQ. Una vez construido, los patrones a clasificar se etiquetarán usando la regla del vecino más próximo tomando como referencia al conjunto P_{LVQ} [Cortijo y Perez de la Blanca, 1997].

Así, el objetivo del aprendizaje adaptativo con la técnica LVQ es la construcción del conjunto de referencia P_{LVQ}. Este proceso puede descomponerse en dos pasos:

1. Inicialización del conjunto de prototipos. El procedimiento recomendado es la selección de prototipos con ciertas restricciones que aseguran que éstos se encuentran dentro del agrupamiento correspondiente a su clase. Se puede asegurar que un prototipo está dentro de un agrupamiento si su clasificación mediante la regla k vecinos más próximos [Geva y Sitte, 1991] es correcta. Esta estrategia de selección acelera la convergencia del aprendizaje.

2. Aprendizaje (o corrección de los parámetros). Consiste en actualizar de forma iterativa los prototipos del conjunto de referencia hasta que se consiga estabilidad o se cumpla algún criterio de convergencia adicional. Ejemplos de funciones de corrección de prototipos son los métodos LVQ1 y O-LVQ.

Algoritmo LVQ1

Se supone que un determinado número de prototipos $\mathbf{m}_i$ son ubicados en el espacio de entrada aproximando la distribución de los patrones $\mathbf{x}(\ell)$ mediante sus valores cuantizados. Usualmente se asignan varios prototipos por cada una de las L clases del conjunto $K = \{k_1, k_2, ..., k_L\}$. Se define el prototipo $\mathbf{m}_i$ más cercano a $\mathbf{x}$, denotado $\mathbf{m}_c$ como

$$c = \arg\min\{\|\mathbf{x} - \mathbf{m}_i\|\}. \tag{1.3}$$

Partiendo de valores de los prototipos $\mathbf{m}_i$ adecuadamente inicializados, las ecuaciones siguientes definen el proceso básico del algoritmo de entrenamiento LVQ1 [Kohonen, 1990]

$$\mathbf{m}_c \leftarrow \mathbf{m}_c + s(\ell)\alpha[\mathbf{x}(\ell) - \mathbf{m}_c], \tag{1.4}$$
$$\mathbf{m}_i \leftarrow \mathbf{m}_i \text{ para } i \neq c, \tag{1.5}$$

donde

$$s(\ell) = \begin{cases} +1 & \text{si } \mathbf{x}(\ell) \text{ y } \mathbf{m}_c \text{ pertenecen a la misma clase,} \\ -1 & \text{si } \mathbf{x}(\ell) \text{ y } \mathbf{m}_c \text{ pertenecen a diferentes clases.} \end{cases}$$

La tasa de aprendizaje $0 < \alpha < 1$ puede ser constante o decrecer monótonamente con cada paso de entrenamiento.

Algoritmo LVQ optimizado

En el caso del LVQ optimizado (O-LVQ) cada uno de los N_p prototipos del conjunto de referencia tiene su propia velocidad de aprendizaje, por lo que en (1.3) se sustituye α por α_i para $i = 1, 2 \ldots N_p$ [Kohonen et al., 1996]:

$$\mathbf{m}_c \leftarrow \mathbf{m}_c + s(\ell)\alpha_c[\mathbf{x}(\ell) - \mathbf{m}_c]. \tag{1.6}$$

La precisión estadística del conjunto de prototipos (en inglés *codebook*) es óptima si los efectos de las correcciones realizadas en distintos pasos de tiempo son de

igual peso. Se puede demostrar que los valores óptimos de α_i en este sentido se obtienen utilizando la expresión:

$$\alpha_c \leftarrow \frac{\alpha_c}{1 + s(\ell)\alpha_c}, \tag{1.7}$$

la cual permite que todos los patrones afecten el entrenamiento por igual y provee una rápida convergencia [Kohonen, 1992]. Teniendo en cuenta la restricción $0 < \alpha_i < 1$ puede verse que $s(\ell)$ hace que α_c decrezca cuando se clasifica correctamente y crezca cuando se clasifica incorrectamente.

1.3.2. Modelos ocultos de Markov

Los modelos ocultos de Markov (HMM, del inglés *hidden Markov models*) son métodos estadísticos originalmente utilizados para modelar series de datos de tiempo discreto. La hipótesis principal de los HMM es que los datos pueden ser caracterizados como un proceso aleatorio paramétrico y los parámetros del proceso estadístico pueden ser estimados de manera precisa [Huang et al., 1990; Jelinek, 1999]. Actualmente los HMM son la herramienta más utilizada para modelar la voz, debido a que han permitido capturar la gran variabilidad temporal de este tipo de señales. También son utilizados para modelar infinidad de tipos de series y secuencias temporales.

Un HMM puede ser considerado una máquina de estados finitos, ya que contiene un conjunto finito de estados que están conectados entre sí mediante arcos de transición con probabilidades asociadas. En cada instante de tiempo el modelo se encuentra en un estado al que llega según una probabilidad de transición que depende del estado anterior. Luego de la transición, el estado emite un símbolo de acuerdo a su distribución de probabilidad sobre el conjunto de símbolos de salida.

En la Figura 1.3, a modo de ejemplo, se puede observar un HMM discreto de cinco estados, donde los estados 1 y 5 no emiten mientras que los estados del 2 al 4 pueden emitir uno de dos símbolos posibles (N y P) en cada instante de tiempo. Este modelo se denomina de izquierda a derecha porque la probabilidad de transición entre un estado s_t a un estado $s_{t+1} < s_t$ es siempre nula. Los parámetros a_{ij} representan las probabilidades de transición del estado i al estado j y los parámetros $b_s(\psi)$ representan la probabilidad de que el estado s emita el símbolo ψ en un instante de tiempo determinado. En el caso de los HMM continuos, las distribuciones de los símbolos observables son densidades de probabilidad definidas sobre el espacio de observación. Cada uno de los estados puede emitir, con

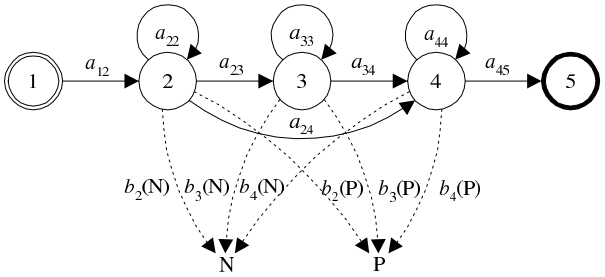


Figura 1.3. Esquema de un HMM de izquierda a derecha y discreto de cinco estados [Milone, 2004].

cierta probabilidad, cualquiera de los símbolos del conjunto de símbolos observables. Por lo tanto, no es posible determinar con certeza la secuencia de estados (o camino) del modelo observando solamente los símbolos de salida, es decir el funcionamiento interno del sistema queda oculto [Rabiner y Juang, 1986].

Descripción formal del modelo

Un HMM continuo se caracteriza por un conjunto de parámetros:

$$\Theta = \langle \mathcal{Q}, \mathbb{O}, \mathbf{A}, \mathcal{B} \rangle, \tag{1.8}$$

siendo:

- $\mathcal{Q}$ el conjunto de estados del modelo,
- $\mathbb{O}$ el espacio de observaciones,
- $\mathbf{A}$ una matriz cuyos elementos son las probabilidades de transición entre estados y
- $\mathcal{B}$ el conjunto las de distribuciones de probabilidad de observación.

Para modelar las funciones de densidad de probabilidad de observación usualmente se utilizan combinaciones lineales o mezclas de distribuciones Gaussianas [Rabiner, 1989]

$$b_j(\mathbf{x}) = \sum_{k=1}^{M} c_{jk} \mathcal{N}(\mathbf{x}, \mathbf{u}_{jk}, \mathbf{\Sigma}_{jk}), \tag{1.9}$$

donde $\mathcal{N}(\mathbf{x}, \mathbf{u}_{jk}, \mathbf{\Sigma}_{jk})$ denota una función de densidad de probabilidad Gaussiana del estado j, con vector de medias $\mathbf{u}_{jk}$ y matriz de covarianzas $\mathbf{\Sigma}_{jk}$ en el estado j, M es el número de componentes en la mezcla y c_{jk} es el peso relativo de la k-ésima componente Gaussiana, que satisface $\sum_{k=1}^{M} c_{jk} = 1 \ \forall j \in \mathcal{Q}$. Los HMM semi-continuos se diferencian en que las componentes Gaussianas son las mismas para todos los estados, variando únicamente los pesos c_{jk} en la mezcla. Esta estrategia de enlazado permite reducir la cantidad de parámetros a estimar en el entrenamiento.

El camino de maximiza probabilidad

Para calcular la probabilidad $P(\mathbf{X}^T|\Theta)$ dada una secuencia de observaciones $\mathbf{X}^T = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T)$ y el modelo Θ, se suman las probabilidades conjuntas para todos los caminos, $\mathbf{q}^T$:

$$P(\mathbf{X}^T|\Theta) = \sum_{\forall \mathbf{q}^T} P(\mathbf{X}^T, \mathbf{q}^T|\Theta) = \sum_{\forall \mathbf{q}^T} P(\mathbf{q}^T|\Theta) P(\mathbf{X}^T|\mathbf{q}^T, \Theta). \tag{1.10}$$

Sin embargo, el cálculo directo de esta ecuación resulta en una complejidad computacional del orden de $2TN^T$, por lo cual no es computacionalmente viable incluso para valores pequeños de N y T. La solución a este problema es el algoritmo *forward* [Rabiner y Juang, 1986], que conserva resultados intermedios para ahorrar cálculos posteriores y reducir la cantidad de operaciones.

Por otro lado, una buena aproximación para $P(\mathbf{X}^T|\Theta)$ es considerar el máximo, es decir la probabilidad dada por el camino más probable, en lugar de la suma sobre todos los caminos posibles:

$$P(\mathbf{X}^T|\Theta) \approx \max_{\forall \mathbf{q}^T} \left\{ P(\mathbf{q}^T|\Theta) P(\mathbf{X}^T|\mathbf{q}^T, \Theta) \right\}. \tag{1.11}$$

De hecho, en la mayoría de los casos es suficiente encontrar el camino más probable para la secuencia dada y su probabilidad asociada. El algoritmo de Viterbi es una técnica basada en programación dinámica que permite encontrar el camino más probable de manera muy eficiente [Viterbi, 1967]. El algoritmo obtiene el camino óptimo que mejor explica la secuencia de observaciones. El algoritmo, en cada tiempo t calcula la probabilidad acumulada:

$$\lambda_t(j) = \max_{\forall \mathbf{q}^{t-1}} \left\{ P(\mathbf{q}^{t-1}, q_t = j, \mathbf{X}^t|\Theta) P(\mathbf{q}^{t-1}|\Theta) \right\} \quad \forall j \in \mathcal{Q}, \tag{1.12}$$

que se puede calcular recursivamente como $\lambda_t(j) = \text{máx}_{\forall i \in \mathcal{Q}}\{\lambda_{t-1}(i)a_{ij}\}b_j(\mathbf{x}_t)$, comenzando con $\lambda_0(j) = 1 \ \forall j \in \mathcal{Q}$. Así, la probabilidad de que el modelo Θ observe la secuencia $\mathbf{X}^T$ se obtiene como:

$$P(\mathbf{X}^T|\Theta) \approx \max_{\forall j \in \mathcal{Q}}\{\lambda_T(j)\}, \tag{1.13}$$

y el camino óptimo $\tilde{\mathbf{q}}^T$ se obtiene de forma recursiva mediante:

$$\tilde{q}_t = \arg\max_{\forall i \in \mathcal{Q}}\{\lambda_t(i)a_{i\tilde{q}_{t+1}}\}, \tag{1.14}$$

para $t = T-1, T-2, ..., 1$, partiendo de $\tilde{q}_T = \arg\text{máx}_{\forall j \in \mathcal{Q}}\{\lambda_T(j)\}$ [Jelinek, 1999].

Entrenamiento del modelo

El principal problema para la aplicación de los HMM es la estimación del conjunto de parámetros del modelo para describir de manera precisa las secuencias de observación de entrenamiento. Este problema se resuelve iterativamente mediante el algoritmo de Baum-Welch, también conocido como *forward-backward*, que puede ser interpretado como una implementación del algoritmo estadístico de maximización de la esperanza [Bahl et al., 1983]. En cada iteración del algoritmo se obtiene una nueva estimación de los parámetros del modelo en base a la estimación de la iteración anterior. Para obtener en cada paso una mejor estimación, se maximiza una función auxiliar basada en la teoría de la información

$$\mathcal{O}(\Theta, \tilde{\Theta}) = \sum_{\forall \mathbf{q}^t} P(\mathbf{X}^T, \mathbf{q}^T|\Theta) \log P(\mathbf{X}^T, \mathbf{q}^T|\tilde{\Theta}), \tag{1.15}$$

para obtener la nueva estimación de los parámetros del modelo $\tilde{\Theta}$ en base a la estimación anterior Θ. A partir de (1.15) se derivan las fórmulas para la reestimación de los parámetros [Milone, 2004]

$$\tilde{a}_{ij} = \frac{\sum_{t=1}^{T} \gamma_t(i,j)}{\sum_{t=1}^{T} \gamma_t(i)}, \tag{1.16}$$

$$\tilde{c}_{jk} = \frac{\sum_{t=1}^{T} \psi_t(j,k)}{\sum_{t=1}^{T} \gamma_t(i)}, \tag{1.17}$$

$$\tilde{\boldsymbol{\mu}}_{jk} = \frac{\sum_{t=1}^{T} \psi_t(j,k)\mathbf{x_t}}{\sum_{t=1}^{T} \psi_t(j,k)}, \tag{1.18}$$

$$\tilde{\boldsymbol{\Sigma}}_{jk}^{-1} = \frac{\sum_{t=1}^{T} \psi_t(j,k)(\mathbf{x_t} - \tilde{\mu}_{jk})(\mathbf{x_t} - \tilde{\mu}_{jk})^{\mathtt{T}}}{\sum_{t=1}^{T} \psi_t(j,k)}, \tag{1.19}$$

donde, por simplificación, se han utilizado las variables γ y ψ definidas a continuación [Milone, 2003]. La variable

$$\gamma_t(i) \triangleq P(q_t = i|\mathbf{X}^T, \Theta)), \tag{1.20}$$

se puede interpretar como la cantidad de veces que el modelo se encuentra en el estado i en tiempo t, observando $\mathbf{X}^T$. Por otro lado, la cantidad de transiciones desde el estado i hacia el estado j, en tiempo t y observando $\mathbf{X}^T$ se representa mediante

$$\gamma_t(i,j) \triangleq P(q_{t-1} = i, q_t = j|\mathbf{X}^T, \Theta). \tag{1.21}$$

Y por último, la variable

$$\psi_t(j,k) \triangleq P(q_t = j, k_t = k|\mathbf{X}^T, \Theta), \tag{1.22}$$

se interpreta como la cantidad esperada de veces que el modelo Θ llegó al estado j en tiempo t, utilizando la Gaussiana k y observando la secuencia $\mathbf{X}^T$. Empleando estas fórmulas se puede mejorar iterativamente la probabilidad del modelo de observar la secuencia $\mathbf{X}^T$ hasta alcanzar un criterio de finalización.

Clasificación de patrones mediante HMM

Un clasificador basado en HMM generalmente consiste en M modelos, siendo M la cantidad de clases a discriminar en un problema dado. Cada uno de estos modelos es entrenado con los patrones de una de las clases, es decir, las distintas clases son modeladas de manera independiente. Por ejemplo, si se utiliza un HMM para construir un clasificador de fonemas, se tienen entonces M fonemas (clases) para ser clasificados y un conjunto de entrenamiento de P_m ocurrencias

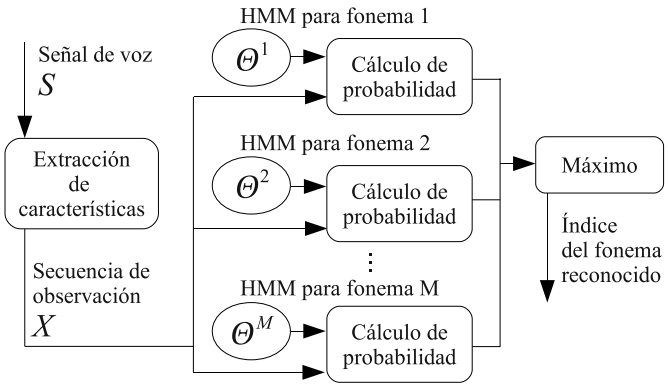


Figura 1.4. Diagrama de un clasificador basado en HMM.

de cada fonema. Cada ocurrencia de un fonema, parametrizada de alguna manera que represente adecuadamente sus características temporales y frecuenciales, constituye una secuencia de observación. En la Figura 1.4 se puede ver el diagrama de un clasificador de fonemas basado en HMM, donde para cada fonema m se tiene un modelo Θ^m cuyos parámetros maximizan la probabilidad de los patrones de entrenamiento correspondientes. Para un patrón desconocido el algoritmo de Viterbi calcula la probabilidad de que éste haya sido generado por cada uno de los M modelos y elige el modelo cuya probabilidad sea máxima.

1.4. Algoritmos evolutivos

La computación evolutiva es una rama de la inteligencia computacional que engloba un amplio conjunto de técnicas de optimización meta-heurística. El término fue introducido recientemente para reunir bajo un común denominador a todos los enfoques que simulan distintos aspectos de la evolución natural.

Todos los organismos vivientes poseen cierta información genética que los caracteriza. Esta información, codificada en genes, les permite transmitir sus características a través de generaciones sucesivas. Dichas características determinan su aptitud para sobrevivir en el entorno donde compiten con sus pares. Debido a la competencia que ocurre entre los individuos, por ejemplo para conseguir alimento o para reproducirse, mientras más apto sea un individuo mayor probabilidad tiene éste de sobrevivir. Es decir, los individuos más aptos tendrán mayor

descendencia que los individuos menos aptos, y éstos últimos tenderán a desaparecer. Todo individuo descendiente combina características de sus progenitores junto con otras características propias, y si la combinación resulta en una buena adaptación a su entorno, el nuevo individuo tendrá su propia descendencia. Luego, a lo largo de las generaciones, cada nueva población será en promedio más apta que la población anterior.

Inspiradas en estas ideas, las distintas técnicas de computación evolutiva son aplicadas para resolver una amplia variedad de problemas de optimización. En los algoritmos de búsqueda y optimización basados en computación evolutiva se tiene una población en la que cada individuo representa una solución diferente. Además, en base al problema que se desea optimizar, se define una función de aptitud para simular un entorno. Luego a cada individuo se le asigna un valor de aptitud, que indica qué tan adaptado está al entorno. Según su aptitud, cada individuo tendrá mayor o menor probabilidad de generar descendencia, la cual se obtiene mediante la aplicación de operadores genéticos.

Las dos características fundamentales que distinguen a estos algoritmos de otros métodos computacionales de búsqueda y optimización que no se basan en la naturaleza son [de los Cobos Silva et al., 2010]:

- Que trabajan con una población de individuos que representan a un conjunto de soluciones del problema.
- Que existe comunicación e intercambio de información entre los individuos de la población.

Dentro de los algoritmos evolutivos (EA, del inglés *evolutionary algorithms*) existen tres variantes principales: los algoritmos genéticos, las estrategias evolutivas y la programación evolutiva. A partir de estos tres enfoques principales se han derivado innumerables variantes cuyas principales diferencias consisten en la forma de representar a los individuos, el diseño de los operadores de variación, y los mecanismos de selección y reproducción [Bäck et al., 1997]. En las siguientes secciones se describen las principales características del enfoque de interés para el desarrollo de la tesis, los algoritmos genéticos.

1.4.1. Optimización

Optimización es el proceso de obtener el mejor resultado posible en las circunstancias dadas. Usualmente, el esfuerzo que se requiere minimizar, o el beneficio que se desea maximizar, puede ser expresado como función de ciertas variables.

Luego, se puede definir optimización como el proceso de encontrar las condiciones bajo las cuales se obtiene el máximo (o el mínimo) de la función objetivo sin enumerar y evaluar todas las alternativas posibles.

Definición 1 *La función $f : S \to \mathbb{R}$, donde $S \subseteq \mathbb{R}^n$, se denomina función objetivo y el conjunto S es el conjunto de soluciones posibles.*

Luego, en el caso de maximización, el problema consiste en encontrar $\mathbf{x}^* \in S$ que satisfaga [Ravindran et al., 2006]

$$f(\mathbf{x}^*) \geq f(\mathbf{x}) \text{ para toda } \mathbf{x} \in S, \tag{1.23}$$

sujeto a ciertas restricciones.

La elección de la función objetivo para un caso particular estará sujeta a la naturaleza del problema. Existen casos en los que la optimización con respecto a un criterio puede conducir a resultados que no son satisfactorios con respecto a otro criterio. Por lo tanto, la elección de la función objetivo puede ser una de las decisiones más importantes en el proceso de optimización [Rao, 2009].

No existe un único método que sea adecuado para abordar cualquier problema de optimización, sino que se han desarrollado una variedad de métodos para resolver distintos tipos de problemas. Las técnicas clásicas son adecuadas para resolver problemas con características específicas, y requieren información que no siempre está disponible. Por ejemplo, los métodos basados en gradiente requieren la primer derivada de la función objetivo, y el método de Newton requiere, además, conocer la segunda derivada. La aplicación de estos métodos está, entonces, limitada a problemas donde la función objetivo sea diferenciable [Ravindran et al., 2006]. Más aún, en muchos problemas del mundo real, como en el problema abordado en esta tesis, la función objetivo no está dada en forma explícita y estos métodos no pueden aplicarse. También existen problemas en los cuales el espacio de búsqueda es tan grande que las técnicas clásicas de búsqueda y optimización requieren tiempo exponencial. En estos casos resulta necesario recurrir a las técnicas heurísticas, que permiten encontrar soluciones buenas en tiempos aceptables. Si bien no se puede garantizar la optimalidad de las soluciones encontradas, en general se obtienen soluciones cercanas al óptimo global [Reeves, 1993].

1.4.2. Algoritmos genéticos

Los algoritmos evolutivos más elementales son los algoritmos genéticos (GA, del inglés *genetic algorithms*), que se hicieron populares a partir de un trabajo de

Algoritmo 1: Pasos de un algoritmo genético convencional.

Generar la población inicial
Evaluar la población
repetir
  Seleccionar padres
  Aplicar operadores genéticos
  Reemplazar la población actual
  Evaluar la población
hasta *alcanzar el criterio de finalización*

John Holland en 1975. Fueron introducidos como un modelo general de proceso adaptativo basado en las leyes del proceso de evolución natural [Goldberg, 1989], pero el mayor campo de aplicación de éstas técnicas es el de la optimización. Los GA se diferencian de los algoritmos de búsqueda aleatoria porque combinan elementos de búsqueda dirigida y estocástica, lo cual los hace más robustos que los algoritmos de búsqueda clásicos [Michalewicz, 1992]. Además, los GA realizan una búsqueda multi-direccional, manteniendo una población de soluciones potenciales, entre las cuales se intercambia información.

En los GA tradicionales, la población consiste en un grupo de individuos cuya información está codificada en cromosomas binarios, es decir, cadenas de bits. De esta manera cada uno de los distintos parámetros del problema (fenotipo) es representado en el cromosoma mediante uno o más bits (genes). Si bien el fundamento teórico de los GA se basa en la codificación binaria, en la práctica este tipo de representación tiene desventajas en ciertas tareas. De hecho, se han realizado experimentos comparando a los GA con otros métodos basados en representaciones de punto flotante y se obtuvo como conclusión que en este último caso se alcanzan soluciones más precisas y en menor tiempo, y a la vez resulta más sencillo incorporar conocimiento específico del problema en el algoritmo [Janikow y Michalewicz, 1991]. En esta sección se describen los GA tradicionales con representación binaria, los operadores de variación más elementales y las bases teóricas de su funcionamiento. Los pasos del algoritmo genético convencional se resumen en el Algoritmo 1. Más adelante, se describirán las representaciones específicas utilizadas y los operadores diseñados para atacar los problemas de optimización planteados en la tesis.

Codificación

Un cromosoma, que representa una potencial solución para el problema, se representa como una cadena de variables en la que cada elemento se denomina gen. El esquema de codificación es un punto clave de cualquier GA porque puede limitar severamente la ventana de información que es observada por el sistema [Koza, 1990]. Por un lado, la codificación binaria utilizada en los GA clásicos brinda la máxima cantidad de esquemas, o cromosomas que comparten características, lo cual facilita al algoritmo la preservación de las combinaciones que resultan en individuos aptos [Goldberg, 1994]. La codificación binaria, además, facilita el análisis teórico de la convergencia de los GA [Goldberg, 1989]. Por otro lado, las representaciones de punto flotante permiten acercar el algoritmo al espacio del problema, incorporando información específica en los operadores. Esto evita realizar un mapeo entre las variables definidas por el problema y la codificación empleada, tarea que no resulta sencilla en la mayoría de los problemas reales [Michalewicz, 1992]. Otra ventaja de las codificaciones de punto flotante (como en las de números enteros) es que dos puntos que están cercanos en el espacio de la representación también están cercanos en el espacio del problema, mientras que no siempre es así en el caso de las codificaciones binarias [Janikow y Michalewicz, 1991]. La representación clásica tiene también la desventaja de que la precisión está limitada por la cantidad de bits con la que se codifican las variables, es decir, por el tamaño del cromosoma. Si bien siempre se puede obtener la precisión deseada con la cantidad necesaria de bits, nunca es deseable extender demasiado la longitud del cromosoma ya que esto hace decrecer la velocidad de convergencia del algoritmo. Se puede concluir, entonces, que existe un compromiso entre la cardinalidad del alfabeto de la codificación y la longitud del cromosoma.

1.4.3. Función objetivo y operadores de selección

Función objetivo

La función objetivo provee el mecanismo para evaluar el estatus de cada cromosoma. La misma toma como entrada un cromosoma y produce un número o una lista de números que indican una medida del desempeño del cromosoma en el problema a resolver. Sin embargo el rango de valores varía de un problema a otro. Para mantener uniformidad sobre varios dominios de problemas se necesita una función para mapear el valor de objetivo a un valor de aptitud. Para ésto existen diferentes métodos llamados técnicas de aptitud. Las dos más utilizadas

son las siguientes:

- Ventaneo: suponiendo que el valor objetivo del peor cromosoma en la población es V_w, a cada cromosoma se le puede asignar un valor de aptitud f_i proporcional a la diferencia entre el cromosoma i y el peor cromosoma. Se expresa de la forma siguiente:

$$f_i = c \pm (V_i - V_w), \tag{1.24}$$

  donde V_i es el valor de objetivo del cromosoma i y c es una constante. Si se trata de un problema de maximización se utiliza un signo positivo en la ecuación (1.24), y en el caso de un problema de minimización se utiliza un signo negativo.

- Normalización lineal: los cromosomas se clasifican en orden ascendente o descendiente de valor objetivo, dependiendo de si la función objetivo va maximizarse o minimizarse. Asignando al mejor cromosoma un valor de aptitud f_b, luego la aptitud del cromosoma i en la lista ordenada es asignada mediante una función lineal:

$$f_i = f_b - (i-1)d, \tag{1.25}$$

  donde d es la razón de reducción. Esta técnica asegura que el valor objetivo medio se mapea como el valor de aptitud medio.

Selección de progenitores

La selección de progenitores emula el mecanismo de supervivencia del más apto en la naturaleza. Se espera que un cromosoma más apto da un número mayor de descendencia y por consiguiente una mayor probabilidad de sobrevivir en la generación siguiente. Existen varias maneras de realizar una selección efectiva, incluyendo jerarquización, torneo y esquema proporcional [Goldberg, 1989; Withley, 1987], pero siempre el asunto clave es dar preferencia a los individuos más aptos sin descartar al resto.

Por ejemplo, en el esquema proporcional, el cromosoma x con valor de aptitud $f(x,t)$ tiene una razón de crecimiento definida como:

$$\vartheta(x,t) = \frac{f(x,t)}{F(t)}, \tag{1.26}$$

donde $F(t)$ es la aptitud promedio de la población. Un algoritmo comúnmente utilizado para implementar el esquema proporcional es de la "rueda de la ruleta" [Goldberg, 1989], que consiste en los siguientes pasos:

- Obtener la aptitud total sumando las aptitudes de todos los miembros de la población.
- Generar un número aleatorio entre 0 y el valor de aptitud total, n.
- Retornar el primer individuo cuyo valor de aptitud sumado a la aptitud de los individuos anteriores a éste en la población es mayor o igual a n.

Otros métodos de selección utilizados son el de "competencias", el de "ventanas" y el de "ranking" [Goldberg, 1994]. En el primer caso para encontrar un progenitor se seleccionan dos o más individuos aleatoriamente y se los hace competir, es decir, se conserva el de mayor aptitud. En el caso de la selección por ventanas se ordena a la población de mayor a menor valor de aptitud y se selecciona aleatoriamente un individuo dentro de una ventana. Dicha ventana está ubicada al comienzo, su tamaño es igual o menor al tamaño de la población y se va reduciendo cada vez que se selecciona un nuevo individuo. Por último, en el método de selección basado en ranking también se ordenan los individuos según su aptitud, pero luego estos valores se actualizan a la posición de cada individuo en el ranking. De esta manera el mejor individuo tendrá aptitud igual al tamaño de la población, mientras que el peor individuo recibirá aptitud igual a 1. Este método puede reducir la velocidad de convergencia, pero tiene como ventaja que todos los cromosomas tienen chance de ser seleccionados.

1.4.4. Reproducción

Operadores genéticos

La cruza es un operador de recombinación que combina partes de dos cromosomas padre para producir una descendencia que contiene material genético de ambos. Para determinar la frecuencia con la que se efectúan los cruzamientos se fija una probabilidad p_c. El operador de cruza es uno de los principales factores que distinguen a los GA de otros algoritmos de optimización. Existen diferentes variantes del operador de cruza y el más simple es el de cruza en un sólo punto, donde se elige un punto de cruzamiento aleatoriamente y las partes de los cromosomas más allá de este punto se intercambian para formar la descendencia.

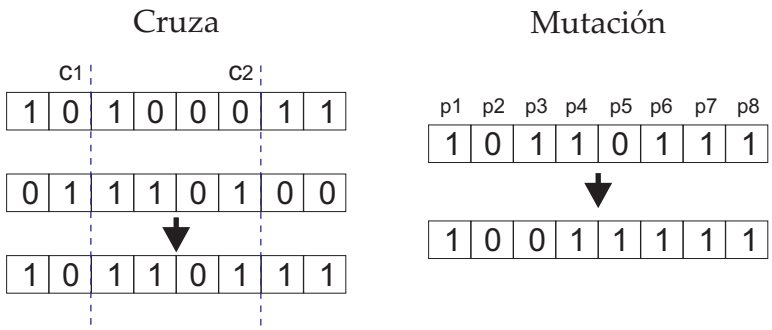


Figura 1.5. Operaciones de cruza y mutación.

La cruza multipunto es similar al cruzamiento de un sólo punto excepto que se eligen m posiciones aleatoriamente. Ambos se pueden ver como una cadena binaria del mismo tamaño que los cromosomas donde en cada posición, un 1 indica que el bit correspondiente debe intercambiarse y un 0 indica que no hay intercambio. En general, para codificaciones no binarias la cruza puede implementarse de manera similar.

La mutación es un operador que introduce variaciones en los cromosomas. Esta variación puede ser global o local. La operación ocurre ocasionalmente con una probabilidad pequeña p_m pero altera aleatoriamente el valor de un gen en un cromosoma. Cada bit de una cadena de bits se reemplaza con un bit generado aleatoriamente si se pasa un test de probabilidad. Otra variante estándar de mutación cambia un 1 por un 0 o un 0 por un 1 si se pasa el test de probabilidad. Este método resulta en una tasa de mutación dos veces más alta que el método anterior.

Estrategias de reemplazo

Pueden proponerse las siguientes estrategias para realizar el reemplazo en la población [Tang et al., 1996]:

- *Reemplazo generacional*: cada población de tamaño n genera un número igual de nuevos cromosomas para reemplazar completamente la población vieja. Esta estrategia puede hacer que el mejor miembro de la población falle en reproducir descendencia en la próxima generación. Entonces el método

es comúnmente combinado con una estrategia elitista donde un cromosoma o algunos de los mejores cromosomas son copiados en la generación siguiente. La estrategia elitista puede incrementar la velocidad de dominación de una población mediante un super-cromosoma, pero en balance mejora el desempeño.

- *Reproducción de estado estable*: Esta estrategia significa que sólo algunos pocos cromosomas son reemplazados para producir la población siguiente. Usualmente son reemplazados los peores cromosomas cuando se insertan cromosomas nuevos en la población.

- *Brecha generacional*: se mantiene una fracción de los individuos de la población vieja para insertarlos en la población nueva.

1.4.5. Base teórica de los algoritmos genéticos

El estudio de la convergencia de los GA se basa en la noción de esquema de Holland [Holland, 1975; Michalewicz, 1992]. Simplemente enuncia que los esquemas son conjuntos de cadenas que tienen una o más características en común. Un esquema es construido introduciendo un símbolo comodín "#" en el alfabeto de genes, y representa todas las cadenas (un hiperplano-plano o subconjunto del conjunto de búsqueda) que coinciden en todas las posiciones que no son "#". Claramente todo esquema se corresponde exactamente con 2^r cadenas, donde r es el número de símbolos comodines.

El efecto de la selección

El valor de aptitud del esquema S en el tiempo t, $f(S,t)$, es determinado como el promedio de las aptitudes de todas cadenas coincidentes en la población. Si se utiliza una selección proporcional en la fase de reproducción, podemos estimar el número de cadenas coincidentes del esquema S en la siguiente generación. La probabilidad de selección es igual a $f(S,t)/F(t)$, donde $F(t)$ es la aptitud promedio de la población actual. El número esperado de ocurrencias de S en la generación siguiente es

$$\xi(S,t+1) = \xi(S,t)\frac{f(S,t)}{F(t)}, \tag{1.27}$$

siendo $\xi(S,t)$ el número de cadenas coincidentes con el esquema S en la generación actual. Siendo

$$\varepsilon = \frac{f(S,t) - F(t)}{F(t)}, \tag{1.28}$$

si $\varepsilon > 0$ significa que el esquema tiene una aptitud por encima del promedio y viceversa. Sustituyendo (1.28) en (1.27) se obtiene que el esquema por encima del promedio recibe un incremento exponencial en el número de cadenas en las siguientes generaciones:

$$\xi(S,t) = \xi(S,0)(1+\varepsilon)^t. \tag{1.29}$$

Efecto del operador de cruza

Durante la evolución de un GA, las operaciones genéticas son disruptivas con el esquema actual; por consiguiente, sus efectos deben ser considerados. Suponiendo que el tamaño del cromosoma es L y se aplica cruzamiento de un solo punto, en general, un punto de cruzamiento se selecciona uniformemente entre $L-1$ posiciones posibles. Esto implica que la probabilidad de destrucción de un esquema S es

$$p_d(S) = \frac{\sigma(S)}{L-1}, \tag{1.30}$$

y la probabilidad de supervivencia del esquema es

$$p_s(S) = \frac{1-\sigma(S)}{L-1}, \tag{1.31}$$

siendo $\sigma(S)$ el tamaño del esquema S, definido como la distancia entre las posiciones fijas exteriores. Este tamaño representa la densidad de información contenida en un esquema. Suponiendo que la probabilidad de cruzamiento es p_c, la probabilidad de supervivencia de un esquema es:

$$p_s(S) \geq 1 - p_c\frac{\sigma(S)}{L-1}. \tag{1.32}$$

Efecto del operador de mutación

Si la probabilidad de mutación de bit es p_m, entonces la probabilidad de supervivencia de un simple bit es $1-p_m$. Definiendo el orden del esquema $o(S)$

como el número de posiciones fijas presentes en el esquema, la probabilidad de que el esquema S sobreviva a una mutación es

$$p_s(S) = (1 - p_m)^{o(S)}. \tag{1.33}$$

Como $p_m \ll 1$, esta probabilidad puede aproximarse mediante

$$p_s(S) = 1 - o(S)p_m. \tag{1.34}$$

Ecuación de crecimiento del esquema

el teorema de los esquemas asegura que la cantidad de buenos individuos se va incrementando con el tiempo de ejecución de un GA. Combinando los efectos de selección, cruzamiento y mutación tenemos una nueva forma para la ecuación de crecimiento del esquema:

$$\xi(S, t+1) \geq \xi(S, t)\frac{f(S,t)}{F(t)}\left[1 - p_c\frac{\sigma(s)}{L-1} - o(S)p_m\right]. \tag{1.35}$$

A partir de esta ecuación puede concluirse que el número esperado de representantes de los esquemas cortos, de bajo orden y aptitud superior al promedio recibe un incremento exponencial en generaciones subsecuentes.

1.4.6. Ventajas de la optimización evolutiva

Los EA son considerados como poderosos optimizadores en muchas áreas. Para explorar la aplicación de EA en el campo de procesamiento de señales es importante introducir algunas de sus más importantes características:

- Robustez: tanto en la teoría como en la práctica los EA han probado que son capaces de proveer una búsqueda robusta en espacios complejos [Tang et al., 1996]. Además, estos algoritmos no están limitados por suposiciones restrictivas acerca del espacio de búsqueda (continuidad, existencia de derivadas, unimodalidad, etc.).

- Multimodalidad: los algoritmos de optimización convencionales que utilizan descenso por gradiente pueden quedarse estancados en mínimos locales. Como se discute en [So et al., 1994], los EA son útiles en este tipo de problemas por su habilidad para escapar de los mínimos locales.

- Múltiples objetivos: han demostrado ser un método poderoso para problemas multi-objetivo [Nicolson y Cheetham, 1993], permitiendo obtener un conjunto de soluciones óptimas en lugar de una única solución.

El poder de los EA proviene del hecho de que se trata de una técnica robusta, y pueden tratar con éxito una gran variedad de problemas provenientes de diferentes áreas, incluyendo aquellos en los que otros métodos encuentran dificultades. Si bien no se garantiza que se encuentre la solución óptima del problema, existe evidencia empírica de que se encuentran soluciones de un nivel aceptable, en un tiempo competitivo con el resto de algoritmos de optimización combinatoria.

Las principales diferencias de los algoritmos genéticos con otros métodos de búsqueda son las siguientes:

- Trabajan con la codificación del conjunto de parámetros, no con los parámetros en sí.
- Buscan a partir de una población de puntos, no un punto único.
- Usan información de una función objetivo (o más), y no necesitan derivadas u otro conocimiento adicional.
- Usan reglas probabilísticas como herramienta, no como guía de la optimización.

Un EA, gracias a sus características intrínsecas de paralelismo, puede ser paralelizado de varias maneras para reducir el tiempo demandado por la optimización [Chipperfield y Fleming, 1994]. Entre los métodos de paralelización de los EA se pueden mencionar: la paralelización global, la migración y la difusión. Estas estrategias reflejan diferentes maneras en que el paralelismo puede explotarse en EA y la naturaleza de la estructura de la población y mecanismos de recombinación usados.

Un EA global trata a la población entera como una única evolución y se basa en la arquitectura maestro-esclavo. Los esclavos se encargan de evaluar los individuos, para realizar la selección de los más aptos. El maestro, en cambio, sólo se encarga de realizar las operaciones genéticas entre los individuos para conformar una nueva población en cada generación. Esta estrategia es la que fue seguida para la paralelización del EA implementado en los experimentos de la tesis. En el Apéndice A se pueden encontrar más detalles de la paralelización del EA.

1.4.7. Algoritmos evolutivos multi-objetivo

Consideremos, como ejemplo sencillo de un problema de optimización, la adquisición de una computadora portátil. En dicho caso un comprador desearía satisfacer varios criterios: minimizar el costo y el peso, maximizar las prestaciones (memoria, almacenamiento, interconectividad, etc.) y la satisfacción de sus gustos en cuanto al diseño, etc. En terminología matemática, las marcas y modelos de computadoras disponibles son las variables de decisión, y el proceso de maximizar y minimizar los diferentes criterios se denomina optimización. Para determinar qué tan bien una computadora determinada satisface los criterios implicados se calcula cierta función objetivo a partir de los valores de las variables de decisión. Este tipo de problemas, donde se requiere optimizar un determinado número k de funciones objetivo simultáneamente, se denominan problemas multi-objetivo (PMO). Los PMO pueden requerir la maximización de los k objetivos, la minimización de los k objetivos, o una combinación de maximización y minimización de estas k funciones.

Una característica importante de este tipo de problemas es que no existe una única solución, sino un conjunto de soluciones. La elección de una solución particular es una decisión de compromiso entre distintos puntos del espacio de soluciones, y se realiza en base a la teoría de optimalidad de Pareto [Ehrgott, 2005]. Además, en general, encontrar el óptimo global de un PMO es un problema NP-completo [de los Cobos Silva et al., 2010].

En los problemas de optimización, por lo general, existen ciertas condiciones que las soluciones aceptables deben satisfacer. En general, existen tres estrategias generales para la solución de los PMO: la más básica que consiste en optimizar únicamente el objetivo de mayor prioridad, la de optimizar en base a alguna función lineal o no lineal que combine a todos los objetivos, y por último la de utilizar un algoritmo multi-objetivo para encontrar el frente de Pareto.

Como los EA pueden codificar soluciones en infinidad de representaciones diferentes, y además son capaces de calcular directamente las funciones objetivo asociadas, ofrecen varias ventajas sobre las técnicas tradicionales para la resolución de problemas multi-objetivo. Estas últimas requieren imponer restricciones y mapeos complejos en el dominio del problema para poder resolverlo [Coello Coello et al., 2007]. Los EA multi-objetivo (MOEA, del inglés *multi-objective evolutionary algorithms*), en general, pueden ser guiados fácilmente por la información del dominio del problema sin necesidad de modificarlo. Por lo que el proceso de búsqueda suele ser sencillo de implementar para una aplicación particular. Si bien

usualmente resulta muy difícil encontrar el frente de Pareto exacto, en general son capaces de encontrar aproximaciones razonablemente buenas en un tiempo computacional aceptable. Los MOEA son capaces de abordar espacios no continuos, no convexos y no lineales, así como problemas cuyas funciones objetivo no son explícitas [Salazar et al., 2006].

Los MOEA se clasifican en dos grupos, el de la primera generación agrupa los primeros EA desarrollados para abordar problemas multi-objetivo (suma ponderada, NSGA [Dias y de Vasconcelos, 2002], NPGA [Erickson et al., 2002], MOGA [Fonseca y Fleming, 1993]), mientras que el grupo de la segunda generación está conformado por los MOEA más recientes y eficientes (SPEA2 [Kim et al., 2004], M-PAES [Knowles y Corne, 2000], PESA [Corne et al., 2001], NSGA-II [Deb et al., 2002]). Las principales características que distinguen a estos últimos de los de la primera generación son: un mecanismo que tiene en cuenta la dominancia de Pareto [Salazar et al., 2006] para la adaptación y una estrategia de elitismo que se basa en el uso de una segunda población para almacenar soluciones no dominadas.

Capítulo 2

Procesamiento de señales de habla

2.1. Introducción

Para poder aplicar con éxito cualquier algoritmo de reconocimiento de patrones es necesario, como paso previo, reducir drásticamente la cantidad de datos de la señal de voz [Bishop, 2007]. Es decir, que la señal debe ser parametrizada de tal manera que se elimine la información irrelevante y al mismo tiempo se conserve toda la información que sea útil para el reconocimiento. Además, esta parametrización debe representar la información relevante en el menor número posible de parámetros.

Como se mencionó en la Sección 1.2.5, la etapa de procesamiento de un sistema de RAH recibe como entrada una señal de voz digitalizada. Como esta señal es de duración arbitraria, el primer paso consiste en dividirla en sucesivos tramos de análisis, para luego procesarlos y obtener un conjunto de parámetros, o vector característico que represente a cada uno de ellos.

El objetivo de este capítulo es exponer algunas nociones básicas sobre la naturaleza del habla y su producción, que son de vital importancia en la búsqueda de una representación óptima. Aquí también se pretende presentar las técnicas de análisis más comunes que permiten obtener diferentes representaciones de la información contenida en una señal. Vale aclarar que esta presentación no es exhaustiva, y sólo se introducen los temas que revisten importancia para el desarrollo de la tesis.

En primer lugar se introducen algunos conceptos relacionados a la señal de voz, su producción y sus características de mayor relevancia para el procesamiento. Se describe la unidad fonética sobre la que se ataca el problema en esta tesis, y se dan más detalles sobre el análisis por tramos. Posteriormente se presenta una

breve introducción de las técnicas de procesamiento que sirven de base para el desarrollo de la tesis. En primer lugar se describe la transformada de Fourier y la transformada de Fourier de tiempo corto, para introducir luego la transformada onditas continua. Posteriormente se presenta la técnica de extracción de características más utilizada actualmente en el RAH, mediante la cual se obtienen los coeficientes cepstrales en escala de mel.

2.2. La señal de voz

2.2.1. Producción de la voz

La señal de voz es una onda sonora consistente en oscilaciones de la presión del aire, que se producen por movimientos de las estructuras anatómicas que conforman el aparato fonador humano. Éste está conformado por los pulmones, la laringe, la faringe, las cavidades oral y nasal y elementos articulatorios como el velo, los dientes, los labios y la lengua. El flujo de aire que da origen a la señal de voz es generado por los pulmones y expulsado a través del tracto vocal, en el cual se encuentra con distintas obstrucciones, dando lugar a distintos tipos de sonido (Figura 2.1). En la laringe se encuentran las cuerdas vocales, compuestas por ligamentos y repliegues membranosos, que actúan directamente sobre el flujo del aire permitiendo un paso continuo o bien generando pulsos. El aparato respiratorio actúa como regulador de parámetros importantes como la energía o intensidad y la división del habla en unidades (sílabas, palabras, etc).

2.2.2. El fonema

El fonema es la menor unidad lingüística desprovista de significado y formada por rasgos distintivos [Quilis, 1993]. Los fonemas son modelos abstractos de sonidos que difieren en su realización acústica, y las distintas variantes de un mismo fonema se las denomina alófonos o simplemente fonos [Rufiner, 2009]. A partir de las distintas configuraciones que puede tomar el tracto vocálico, dando lugar a características de resonancia particulares, y los dos tipos de excitación explicados anteriormente, se le da a los sonidos distintas cualidades fonéticas. Es decir, cada una de las distintas configuraciones que toma el tracto vocal durante una elocución corresponde a un fonema. Estas configuraciones varían de manera constante, por la naturaleza de los órganos que intervienen, y existen estados intermedios que dan lugar a períodos de transición entre fonemas consecutivos,

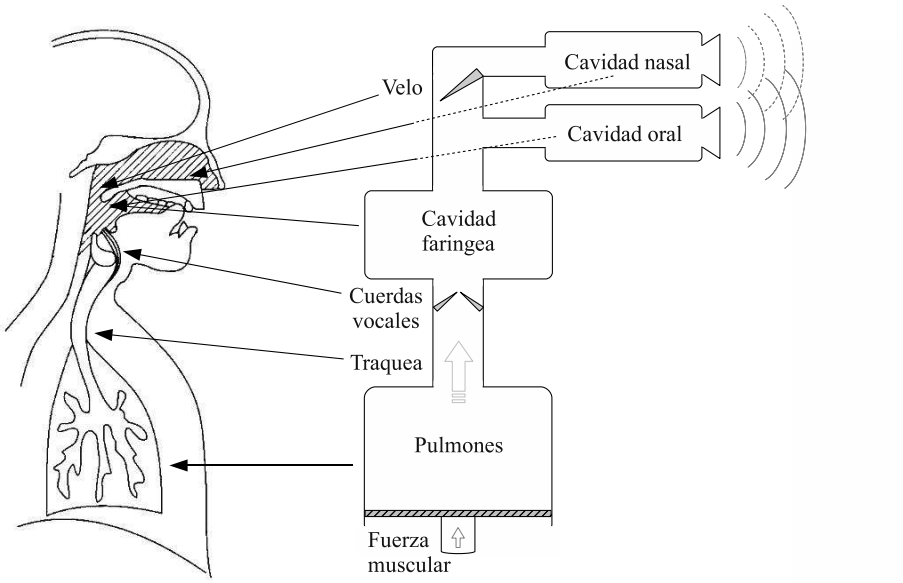


Figura 2.1. Diagrama esquemático (izquierda) y diagrama en bloques (derecha) ilustrando el funcionamiento del aparato fonador humano.

denominados coarticulaciones. Esta es una de las principales dificultades en la tarea de reconocimiento automático, ya que la pronunciación de un fonema depende del contexto, es decir, de los fonema inmediatos anterior y posterior.

Los fonemas pertenecen a una determinada lengua, por ejemplo en los distintos dialectos del español se distinguen hasta 29 unidades diferentes, mientras que en el inglés existen alrededor de 42 fonemas [Deller et al., 1993; Quilis, 1993].

Pensando en el tracto vocal como un sistema, la primer clasificación de los fonemas se realiza en base a al tipo de entrada que lo excita para producir el sonido:

- Los fonemas sonoros se producen cuando las cuerdas vocales vibran, generando pulsos cuasi-periódicos que sirven como excitación del tracto vocal. Ejemplos de sonidos sonoros son las vocales y consonantes como la `/l/` y la `/n/` [Quilis, 1993; Stevens, 2000]. En los hombres, la frecuencia de vibración de las cuerdas vocales varía entre 100 y 170 Hz, en las mujeres entre 180 y 280 Hz, mientras que en los niños puede superar los 300 Hz [Rufiner, 2009]. Esta frecuencia varía en el transcurso de una elocución, de acuerdo al sentido o intención del mensaje y es denominada frecuencia fundamental o entonación (denotada como F_0).

- Los fonemas sordos se producen cuando las cuerdas vocales no vibran, permitiendo el paso ininterrumpido del flujo de aire que se encuentra con obstrucciones en el tracto vocal y produce turbulencias. Este es el caso, por ejemplo, de las consonantes `/s/` y `/f/`.

El tracto vocal se comporta como un filtro, pudiendo variar su configuración de manera continua, y actuar como modulador del flujo de aire permaneciendo relativamente abierto o bien cerrar el paso en un punto específico. La forma del tracto está controlada principalmente por la posición de la lengua, la mandíbula y los labios.

Vocales

Como se mencionó anteriormente, el tracto actúa como un filtro, y en el caso de los sonidos vocálicos, modula los pulsos glóticos para determinar el timbre característico. En el caso de las vocales y los sonidos vocálicos, por ejemplo, el tracto presenta una configuración relativamente abierta y las propiedades de estos sonidos cambian lentamente. Las vocales pueden variar significativamente en duración y por lo general presentan mayor amplitud que el resto de los fonemas. En la Figura 2.2 se puede observar las formas de onda de las vocales del español junto con sus correspondientes espectros. En el dominio temporal se puede apreciar que las formas de onda de las vocales son cuasi-periódicas, debido a que la excitación es un tren de pulsos generado por el movimiento cuasi-cíclico de las cuerdas vocales. En los espectros se pueden apreciar claramente ciertos picos, que indican las frecuencias de resonancia del tracto vocal en la configuración correspondiente a cada fonema. Estas frecuencias de resonancia se denominan formantes, y particularmente las primeras dos formantes (denotadas como F_1 y F_2 respectivamente) son comúnmente utilizadas para caracterizar a las vocales. En la Figura 2.3 se puede observar la distribución de las cinco vocales del español según las primeras dos formantes. Como se puede apreciar, a partir de los valores de F_1 y F_2 las vocales pueden ser fácilmente discriminadas cuando son pronunciadas en forma aislada. Sin embargo, en discurso continuo las clases no se encuentran tan bien separadas y resulta necesario un análisis más complejo.

Según la zona del tracto vocal donde se produce el estrechamiento, las vocales se clasifican en anteriores (`/i/` y `/e/`), medias (`/a/`) y posteriores (`/o/` y `/u/`). Por otro lado, según la abertura de la boca, se clasifican en abiertas (`/a/`), medias (`/e/` y `/o/`) y cerradas (`/i/` y `/u/`).

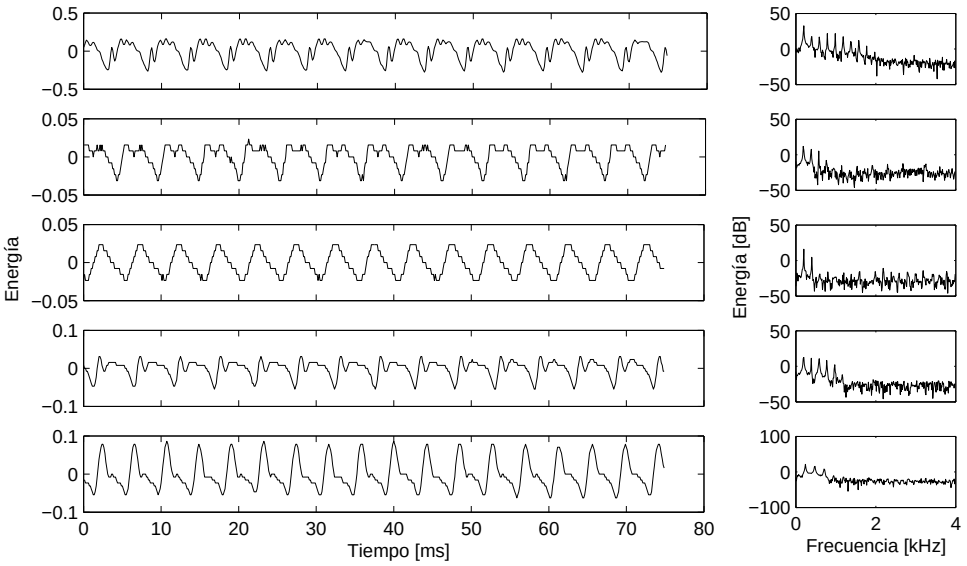


Figura 2.2. Forma de onda temporal (izquierda) y espectros (derecha) de las cinco vocales del español pronunciadas fuera de contexto y sostenidas. Desde arriba hacia abajo: `/a/, /e/, /i/, /o/ y /u/`.

Consonantes

La producción de sonidos consonánticos generalmente envuelve mayores constricciones en el tracto vocal que el caso de las vocales. Las consonantes pueden incluir cualquiera de los dos tipos de excitación mencionados. A diferencia de las vocales, que se producen con una configuración estática del tracto vocal, algunas consonantes requieren un movimiento preciso de los articuladores [Deller et al., 1993].

Según la participación de las cuerdas vocales, el tipo de obstrucción o estrechamiento, y la cavidad por donde se expulsa el aire (bucal o nasal), las consonantes se clasifican en:

- Fricativas: el sonido se forma por medio de un estrechamiento de de los órganos articulatorios, sin que estos lleguen a juntarse. Por ejemplo, las consonantes `/f/` y `/s/`.
- Oclusivas o plosivas: existe un cierre completo de los órganos articulatorios que es luego liberado de forma abrupta. Por ejemplo `/p/`, `/t/` y `/k/`.

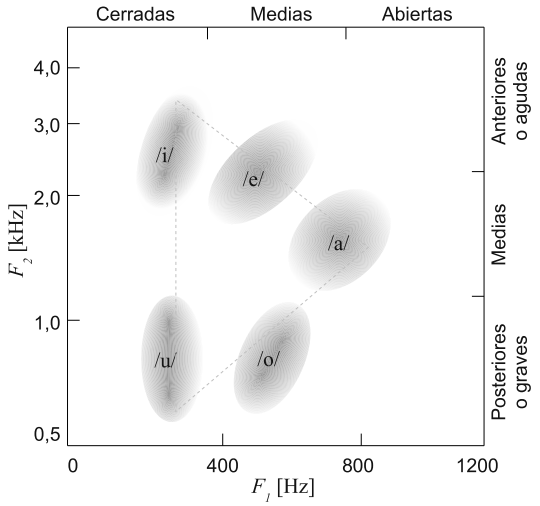


Figura 2.3. Mapa de las formantes F_1 (en escala logarítmica) y F_2 (en escala lineal) para las cinco vocales del español [Milone, 2003]. Sobre el eje F_1 se indica la clasificación según la abertura de la boca y sobre el eje F_2 se indican las zonas de estrechamiento del tracto vocal.

- Nasales: la cavidad bucal está completamente cerrada y el flujo de aire se expulsa exclusivamente por la cavidad nasal. Estas son las consonantes `/m/` y `/n/`.

- Africadas o semi-oclusivas: cuentan con un momento de oclusión seguido de un momento de fricación, como en el caso de `/ch/`.

- Líquidas: en este grupo las cuerdas vocales vibran, y comprende a las laterales, en las cuales el aire se expulsa por los costados de la lengua (`/l/`); y a las vibrantes, que son producidas por vibraciones de la punta de la lengua (`/r/`).

2.3. Análisis de la señal de voz

Algunas señales, como las de habla, se miden en función del tiempo, es decir, en la naturaleza se encuentran en el dominio temporal. Esta representación suele no ser la más apropiada cuando el objetivo es el reconocimiento. En muchos casos, la mayoría de la información discriminativa se encuentra oculta en el contenido

frecuencial de la señal (por ejemplo en las frecuencias formantes). El procesamiento clásico de las señales de voz se ha basado en el análisis por tramos utilizando la transformada de Fourier de tiempo corto [Oppenheim y Schafer, 1989; Rabiner y Schafer, 1978]. Otros métodos de procesamiento de señales de voz que se basan en el análisis por tramos son los coeficientes de predicción lineal [Makhoul, 1975a], el análisis por bandas y el análisis cepstral [Lippmann, 1997]. Sin embargo, la transformada de Fourier de tiempo corto posee limitaciones para el análisis de señales con componentes transitorias debido a su resolución tiempo-frecuencia fija. Un método desarrollado más recientemente es la transformada onditas que permite realizar el análisis de señales no estacionarias en forma más eficiente. La principal ventaja es que realiza un análisis con resolución tiempo-frecuencia variable.

En esta sección se presenta el análisis de la transformada de Fourier y se describen sus características principales. Seguidamente se analiza el problema de la resolución temporal-frecuencial y se introduce la transformada de Fourier de tiempo corto. Una vez mostrados los inconvenientes del análisis tiempo-frecuencia clásico se procede a introducir la transformada onditas continua.

2.3.1. Transformada de Fourier

El análisis ideado por Fourier a principios del siglo XIX descompone una señal en exponenciales de diferentes frecuencias. Desde la perspectiva del procesamiento de señales, se puede ver a este análisis como la técnica de transformar una señal llevándola de una base temporal a una base frecuencial. Estas transformaciones son útiles para el análisis de señales estacionarias, es decir, aquellas cuyas propiedades estadísticas no cambian con el tiempo. La transformada de Fourier (FT, del inglés *Fourier transform*) de una señal $x(t)$ se calcula como [Fourier, 1888]

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-2j\pi ft} dt. \tag{2.1}$$

Los coeficientes de análisis $X(f)$ definen la noción de frecuencia global en una señal. Como se muestra en (2.1), éstos se calculan como productos internos de la señal con las exponenciales la base. Desde un punto de vista más general, se puede pensar en este proceso la comparación de la señal $x(t)$ con un diccionario de señales exponenciales.

Una característica importante de las exponenciales de éste diccionario es que son de duración infinita, y resulta entonces que el análisis de Fourier es útil si las componentes de $x(t)$ son estacionarias. En cambio, si la señal no es estacionaria los

cambios transitorios se esparcen sobre todo el eje de frecuencias en $X(f)$. Es decir, al transformar la señal al dominio de la frecuencia se pierde toda la información temporal, por lo que es imposible saber cuándo toma lugar un evento particular. Es por ello que para el análisis de señales estacionarias por tramos se requiere más que la FT.

2.3.2. Transformada de Fourier de tiempo corto

En un esfuerzo para corregir la deficiencia de la FT, Denis Gabor (1946) adaptó la transformada de Fourier introduciendo un parámetro de "frecuencia local" (local en el tiempo), de tal forma que la transformada de Fourier local analiza sólo una pequeña porción de la señal a la vez. Para ésto, observa a la señal a través de una ventana Gaussiana sobre la cual ésta es aproximadamente estacionaria. Basada en esta adaptación, la transformada de Fourier de tiempo corto (STFT, del inglés *short time Fourier transform*) utiliza una función de ventana localizada temporalmente para realizar un mapeo de la señal en una función bidimensional de tiempo y frecuencia (τ, f):

$$STFT\{x(t)\} = X(\tau, f) = \int_{-\infty}^{\infty} x(t)g(t-\tau)e^{-2j\pi ft}dt. \tag{2.2}$$

$X(\tau, f)$ es simplemente la FT de $x(t)g(t-\tau)$, una función compleja que representa la fase y la magnitud de la señal en función de tiempo y frecuencia. El análisis depende de la elección de la ventana, y sólo puede garantizarse la reconstrucción si $g(t) \in L^2(\mathbb{R})$. Usualmente la ventana es una campana de Gauss, y en este caso los átomos $g_{\tau,f}(t) = g(t-\tau)e^{2j\pi ft}$ son llamados funciones de Gabor. De forma similar a la FT, la STFT se puede pensar como un proceso en el cual se realiza el producto interno de la señal con un diccionario de funciones $g_{\tau,f}(t)$, que en este caso están localizadas en tiempo y en frecuencia.

Si Δf es el ancho de banda del filtro para una ventana $g(t)$ dada, dos sinusoides pueden discriminarse sólo si sus frecuencias están más separadas que Δf, definiendo así a la resolución en frecuencia de la STFT. De manera similar, siendo Δt el ancho de $g(t)$ en el tiempo, dos pulsos pueden discriminarse sólo si están más lejos que Δt. Sin embargo, ni la resolución temporal, ni la frecuencial pueden ser arbitrariamente pequeñas, porque su producto debe cumplir la relación conocida como principio de incertidumbre de Heisenberg:

$$\Delta t \cdot \Delta f \geq \frac{1}{4\pi}. \tag{2.3}$$

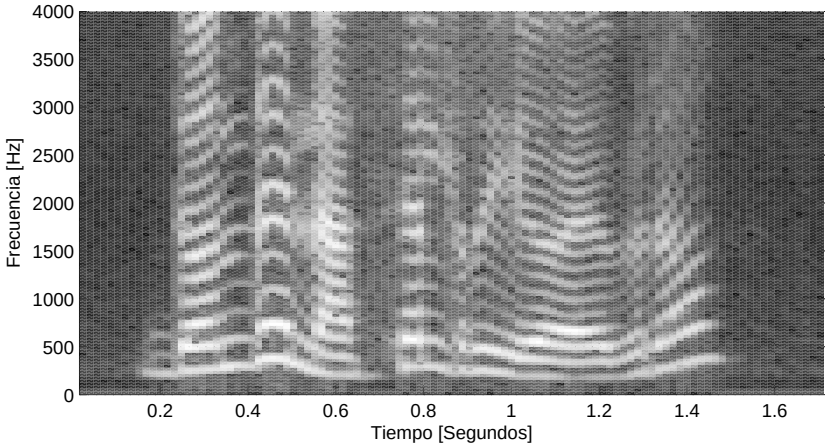


Figura 2.4. Espectrograma de la frase "¿Dónde nace el río Ebro?" del corpus Albayzin [Moreno et al., 1993].

La STFT presenta un compromiso entre dos perspectivas de la señal, la de base en el tiempo y la de base en frecuencia. Provee información sobre cuándo y a qué frecuencias ocurre un evento en la señal, pero con cierta incerteza o incertidumbre. El tamaño de la ventana utilizada determina la incertidumbre del análisis tanto en el dominio del tiempo como en el de la frecuencia. Dicha información generalmente se representa en un espectrograma como el de la Figura 2.4, donde las energías de los espectros de magnitud para cada ventana se mapean en una escala de colores. La desventaja de la STFT es que una vez elegido el tamaño para la ventana temporal, esto fija también el ancho de la ventana para todas las frecuencias. Muchas señales requieren una estrategia más flexible, una en la cuál sea posible variar el tamaño de la ventana para disponer de mayor precisión en el tiempo o en la frecuencia en distintas zonas del plano tiempo-frecuencia.

2.3.3. Transformada ondita continua

Una alternativa a la transformada de Gabor (Sección 2.3.2) es utilizar ventanas moduladas pero de dimensión variable. Más precisamente, la idea es hacer que Δt y Δf cambien en el plano tiempo-frecuencia para obtener un análisis con resolución variable (o multi-resolución). Una manera de producir esto y cumplir con el principio de incertidumbre es hacer que la resolución en el tiempo se

incremente con la frecuencia central de los filtros de análisis. Es decir, el análisis multi-resolución puede estar diseñado para proporcionar una alta resolución temporal (y baja frecuencial) para las frecuencias altas y una alta resolución frecuencial (y baja temporal) para frecuencias bajas.

La transformada ondita continua (CWT, del inglés *continuous wavelet transform*) sigue las ideas anteriores con la simplificación de que las respuestas al impulso de todos los filtros son definidas como versiones escaladas de una misma ondita madre.

Una ondita puede definirse de forma simplificada como una función con valor medio igual a cero, norma unitaria y centrada en la vecindad de 0 [Mallat, 1999]

$$\psi(t) \in L^{\mathbb{R}}, \quad \int_{-\infty}^{\infty} \psi(t)dt = 0, \quad \big\|\psi(t)\big\| = 1. \tag{2.4}$$

A partir de la ondita madre, se obtienen por escalado y traslación los átomos tiempo-frecuencia o simplemente onditas

$$\psi_{s,u}(t) = \frac{1}{\sqrt{|s|}}\psi\left(\frac{t-u}{s}\right), \tag{2.5}$$

donde $s \neq 0$ y u son los parámetros de *escala* y de *traslación*. La función $\psi(t)$ debe verificar además las condiciones de estar bien localizada en tiempo y que su transformada $\widehat{\psi}(\omega)$ sea un filtro continuo pasa-banda, con rápido decaimiento hacia el infinito y hacia $\omega = 0$. Entonces, dada una señal $x(t)$ de energía finita, la CWT de $x \in \mathbf{L}^2(\mathbb{R})$ en el tiempo u y escala s se define como [Mallat, 1989]

$$W_\psi x(s,u) = \langle x, \psi_{s,u}\rangle = \int_{-\infty}^{\infty} x(t)\psi_{s,u}(t)dt. \tag{2.6}$$

La frecuencia local, que depende de la ondita madre, está íntimamente relacionada con las escalas temporales. A diferencia del caso de la STFT, donde se determina por la frecuencia de modulación [Rufiner, 2005]. En la Figura 2.5 se pueden observar algunas de las onditas madres más utilizadas.

A partir del análisis mediante onditas se obtiene en un conjunto de coeficientes que indican cuánto tiene la señal de cada función de la base. La fórmula de reconstrucción está dada por:

$$x(t) = \frac{1}{C_\psi}\int_0^{\infty}\int_{-\infty}^{\infty} W_\psi x(s,u)\psi_{s,u}(t)\frac{\partial u \partial s}{s^2}, \tag{2.7}$$

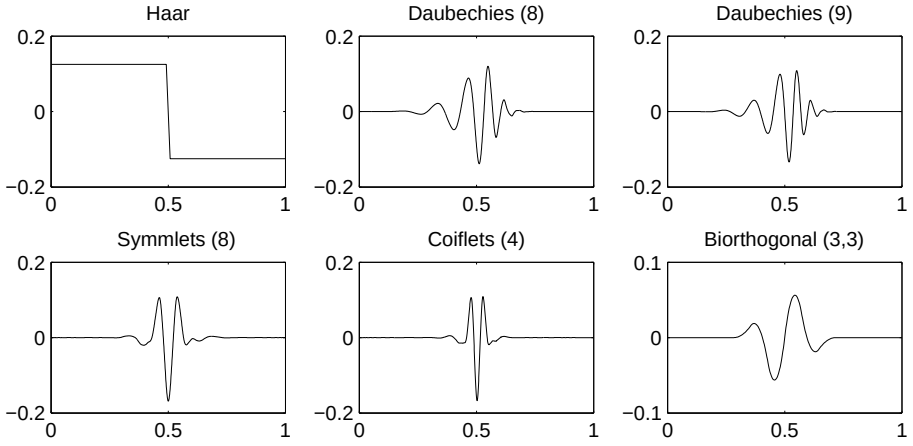


Figura 2.5. Algunas de las onditas madres más difundidas.

donde C_ψ es una constante positiva y con la condición de que $\psi(t)$ sea de energía finita y pasa-banda. Esto implica que $\psi(t)$ oscila en el tiempo como una onda de corta duración y de ahí el nombre de ondita.

La función ondita ψ puede interpretarse como la respuesta al impulso de un filtro pasa-altos y permite obtener, por lo tanto, los denominados coeficientes de detalle de la descomposición. De la misma manera, para obtener las aproximaciones, es decir los coeficientes de bajas frecuencias, se necesita un filtro pasa-bajos [Hess-Nielsen y Wickerhouser, 1996].

Función de escala: cuando $W_\psi x(s,u)$ se conoce sólo para $s < s_0$, para recuperar x se necesita información complementaria correspondiente a $W_\psi x(s,u)$ para $s > s_0$. Esta información es obtenida introduciendo una función de escala ϕ que es una agregación de onditas a escalas mayores que 1. El módulo de su transformada de Fourier se define como

$$|\hat{\phi}(\omega)|^2 = \int_1^{+\infty} |\hat{\psi}(s\omega)|^2 \frac{\partial s}{s}, \tag{2.8}$$

y la fase compleja de $\hat{\phi}(\omega)$ puede ser elegida arbitrariamente. Esta función de escala es justamente la respuesta al impulso del filtro pasa-bajos mencionado. La

aproximación a bajas frecuencias de x en la escala s es

$$L_\phi x(s,u) = \left\langle x(t), \frac{1}{\sqrt{s}}\phi\left(\frac{t-u}{s}\right)\right\rangle. \tag{2.9}$$

Por último, puede demostrarse que $x(t)$ puede reconstruirse mediante [Mallat, 1999]

$$x(t) = \frac{1}{C_\psi}\int_0^{s_0} W_\psi x(s,u) * \psi_{s,u}(t)\frac{\partial s}{s^2} + \frac{1}{C_\psi s_0} L_\phi x(s_0,u) * \phi_{s,u}(t), \tag{2.10}$$

donde $*$ denota convolución continua.

La CWT representa una alternativa a la transformada de Fourier por ventanas que despliega la información de la señal en un mapeo tiempo-frecuencia radicalmente diferente. Otra propiedad relevante de la transformada continua es su invarianza respecto de las transformaciones o cambios de escala de la señal. Estructuras similares serán detectadas de la misma forma, independientemente de su localización en tiempo o escala.

2.3.4. Análisis por tramos

Como se mencionó anteriormente, la forma del tracto vocal varía en el transcurso de una elocución, luego, las características frecuenciales de la señal de voz varían en función del tiempo. Para aplicar técnicas de análisis como por ejemplo el análisis espectral, se requiere que las características de la señal no cambien a lo largo del tiempo. Resulta necesario, por lo tanto, descomponer las señales en intervalos donde sus propiedades se mantengan constantes para realizar el estudio bajo la hipótesis de estacionariedad. De esta manera, es posible capturar las características transitorias de una señal. En el caso de las señales de habla, la velocidad de cambio de la propiedades de la señal está dada por la velocidad con la que el tracto vocal puede cambiar su morfología de manera significativa. Usualmente, en la práctica se analizan las señales de voz en intervalos de 10 a 30 milisegundos de duración, donde se puede suponer que la señal es localmente estacionaria.

De aquí en adelante se supone que la señal de voz de tiempo discreto $s(n)$ se obtuvo mediante un proceso de muestreo uniforme con período T_m a partir de un señal de tiempo continuo $s(t)$. Un tramo de voz se define como el producto entre una ventana desplazada temporalmente y una secuencia de voz:

$$v(n;i) = w(n;N_w)s(n+iN_p), \quad 0 < n \leq N_w, \tag{2.11}$$

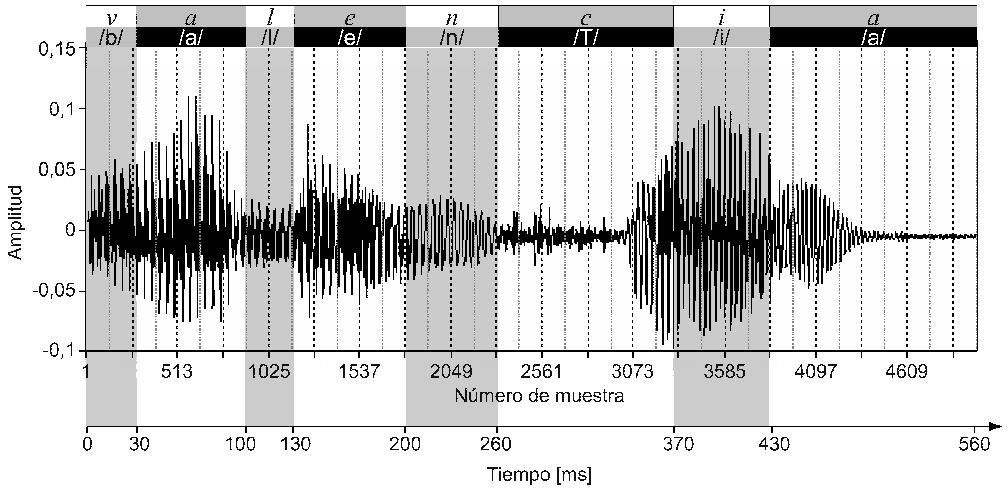


Figura 2.6. Segmentación de una señal de voz en tramos de 256 muestras con solapamiento de 128. En la parte superior se indican los fonemas pronunciados.

siendo i el índice del tramo, N_w el tamaño de la ventana y N_p el paso del análisis por tramos. La señal $v(n;i)$ es un trozo de la señal $s(n)$ cortado con la ventana $w(n;N_w)$, definida para $0 < n \leq N_w$. El tamaño de la ventana adecuado para el análisis depende de las características señal bajo estudio. En el caso de las señales de voz, N_w se determina teniendo en cuenta las consideraciones mencionadas anteriormente. Además, hay que tener en cuenta que al reducir el tamaño de la ventana se mejora la resolución temporal pero se empobrece la resolución espectral. Sin embargo, una estrategia utilizada comúnmente para mejorar la resolución temporal sin este efecto negativo es utilizar un paso menor al tamaño de la ventana. En la práctica habitualmente se utiliza un paso igual a la mitad del tamaño de la ventana, es decir, que dos tramos consecutivos se solapan en la mitad de su longitud.

En cuanto a la forma de la ventana $w(n;N_w)$, existen dos consideraciones a tener en cuenta: por un lado la necesidad de alterar lo menos posible la forma de la señal temporal, y por otro, el requisito de minimizar los artefactos que el ventaneo produce en el espectro. Es decir, el uso de una ventana rectangular permite mantener sin alteración los valores de la señal en el dominio temporal, sin embargo, introduce discontinuidades abruptas en los bordes que provocan distorsiones en el dominio frecuencial. Por esta razón, para preservar las características frecuenciales de la señal, generalmente se utilizan ventanas más suaves como las de Hamming, Hanning o Blackman [Deller et al., 1993]. En la Figura 2.6 se puede

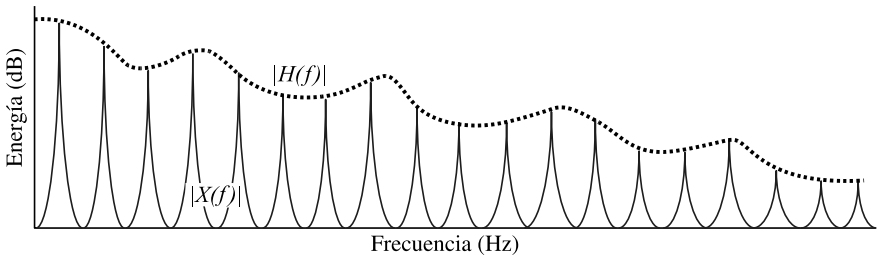


Figura 2.7. Espectro de magnitud representativo de un fonema sonoro simulado. La linea continua corresponde a la señal de excitación $X(f)$ y la linea de puntos a la respuesta del tracto vocal $H(f)$ modulada.

observar un esquema ilustrando la segmentación en tramos de una señal de voz cuya transcripción es la palabra "Valencia". Se indica la duración aproximada de cada fonema en milisegundos y los límites de cada uno de los tramos de 256 muestras. Como se puede apreciar, en este caso el fonema `/a/` es un ejemplo de la variabilidad en la duración de las distintas ocurrencias. Cada uno de los tramos es etiquetado de acuerdo al fonema que predomine entre sus límites y se le aplica una transformación o procesamiento para obtener algún tipo de parametrización que facilite el análisis posterior.

En lo que resta del capítulo, para simplificar la notación, se omitirá el índice i en $v(n;i)$ y con $v(n)$ se hará referencia a un tramo de voz en el cual se supone que la señal es localmente estacionaria.

2.4. Representaciones específicas para el habla

2.4.1. Coeficientes cepstrales

En el modelo de producción de la voz utilizado comúnmente en RAH se supone que la señal es la salida de un sistema lineal. Esto quiere decir que la señal de voz se plantea como resultado de la convolución entre una señal de excitación y la respuesta al impulso del modelo del tracto vocal

$$y(t) = x(t) * h(t). \tag{2.12}$$

En general sólo se conoce $y(t)$, y frecuentemente es deseable obtener sus componentes por separado para poder estudiar las características de la respuesta al impulso del tracto vocal $h(t)$. El análisis *cepstral* resuelve este problema teniendo

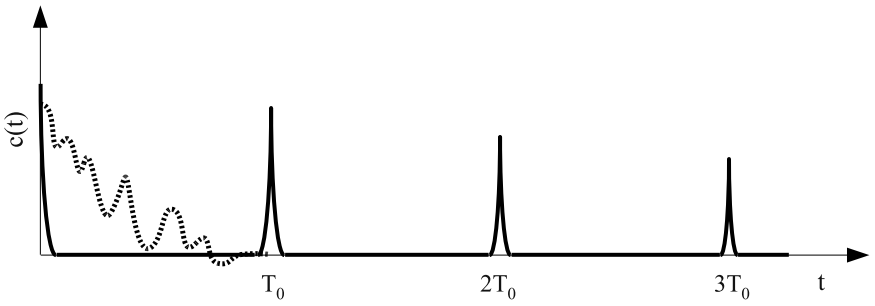


Figura 2.8. Cepstrum representativo de un fonema sonoro simulado. La linea de puntos corresponde a la respuesta del tracto vocal, la linea continua a la señal de excitación y T_0 es el período de entonación.

en cuenta que aplicando la FT a ambos lados de (2.12) se obtiene un producto en el dominio frecuencial,

$$Y(f) = X(f)H(f), \tag{2.13}$$

donde $X(f)$ es el espectro de la excitación y $H(f)$ es la respuesta en frecuencias del tracto vocal. En situaciones no ideales, la señal de habla está dominada por ruido en ciertos rangos frecuenciales, lo cual resulta en una fase que cambia drásticamente de un segmento al siguiente [Huang et al., 2001]. Por ésto, el *cepstrum* complejo es rara vez utilizado y se calcula entonces la magnitud de (2.13), y se aplica luego el logaritmo para obtener una suma:

$$\log_e |Y(f)| = \log_e |X(f)| + \log_e |H(f)|, \tag{2.14}$$

y el *cepstrum* real $c(t)$ de una señal $y(t)$ se calcula como:

$$c(t) = IFT\{\log_e |FT\{y(t)\}|\}, \tag{2.15}$$

donde IFT es la FT inversa.

Los términos *cepstrum* y *cepstral* fueron derivados intercambiando el orden de las letras de las palabras del inglés *spectrum* y *spectral*, para hacer referencia este nuevo dominio.

Esta técnica de procesamiento homomórfico es útil en el RAH porque la velocidad de cambio del espectro de la señal de excitación $X(f)$ es mayor a la velocidad de cambio de la respuesta en frecuencias del tracto vocal $H(f)$ (Figura 2.7). Gracias a esta propiedad de las señales de voz, la señal de excitación y la respuesta del tracto vocal ocupan diferentes zonas en el dominio cepstral,

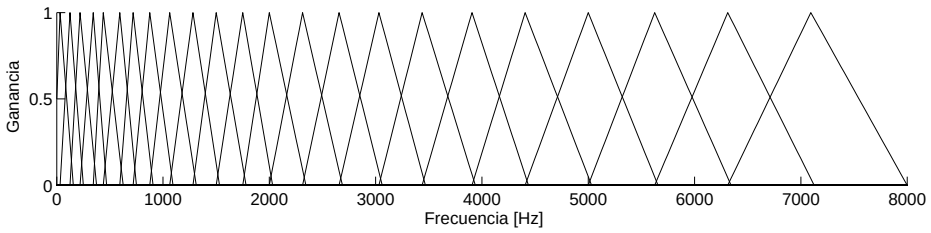


Figura 2.9. Banco de filtros en escala de mel en el rango de frecuencias de 0 a 8 kHz.

permitiendo su separación (Figura 2.8). Esto es muy útil para el reconocimiento porque toda la información que permite la discriminación entre los distintos fonemas está contenida en la respuesta del tracto vocal.

2.4.2. Coeficientes cepstrales en escala de mel

Desde hace tiempo, los coeficientes cepstrales en escala de mel (MFCC, del inglés *mel-frequency cepstral coefficients*) son la representación más utilizada para el RAH [Davis y Mermelstein, 1980]. Esto es así principalmente porque esta representación es adecuada para el uso de HMM, ya que el proceso de estimación de parámetros supone que los vectores de características son no correlacionados. Por otro lado, los MFCC proveen mayor robustez al ruido en comparación con otras técnicas de extracción de características basadas en predicción lineal [Jankowski et al., 1995].

La escala de mel aproxima la relación entre la frecuencia percibida por el oído humano y la frecuencia real. Esta escala perceptual fue obtenida mediante experimentos con tonos puros, en los que a un oyente se le pedía que ajuste la frecuencia de un tono estímulo para que sea la mitad de alto que un tono de referencia [Huang et al., 2001; Rabiner y Juang, 1993]. La relación entre la frecuencia real en Hertz y la frecuencia percibida en *mels* se aproxima mediante:

$$F_{mel}(F_{Hz}) = 2595\ \log_{10}\left(1 + \frac{F_{Hz}}{700}\right). \tag{2.16}$$

Para combinar las propiedades del *cepstrum* y los resultados acerca de la percepción de tonos puros en el ser humano, el espectro de la señal se descompone en bandas de acuerdo a la escala de mel. A partir de esta escala se define el banco de filtros en escala de mel (MFB, del inglés *mel-scaled filter-bank*), compuesto

por ventanas triangulares solapadas. Como se puede observar en la Figura 2.9, el inicio y fin de cada filtro está determinado por la frecuencia central de sus dos filtros adyacentes. El ancho de banda de cada filtro es determinado por las frecuencias centrales, que dependen de la frecuencia de muestreo y de la cantidad de filtros. Es decir, si se incrementa la cantidad de filtros, el ancho de banda de cada filtro decrece. Davis y Mermelstein [Davis y Mermelstein, 1980] propusieron descomponer el espectro de magnitud de una señal de voz de acuerdo a las bandas frecuenciales definidas según el MFB. Más precisamente, en cada banda del espectro de magnitud los coeficientes son escalados por el filtro correspondiente y mediante la integración de éstos se obtiene, como salida, la energía. Los MFCC se calculan aplicando la IFT a la secuencia de los coeficientes de energía obtenidos [Deller et al., 1993]. Sin embargo, como el argumento de la IFT es una secuencia real y par, el cálculo de los MFCC usualmente es simplificado empleando la transformada coseno (CT, del inglés *cosine transform*). Según estas definiciones, los MFCC para un tramo $v(n)$ de una señal de voz digitalizada pueden calcularse mediante

$$\begin{aligned} C_{mel}(m) &= \frac{1}{N_I}\sum_{i=2}^{N_I}\left\{\sum_{k=B_{i-1}}^{k=B_{i+1}}\omega_T\left(k-B_{i-1},B_{i+1}-B_{i-1}\right)\right. \\ &\qquad \left.\times\log_e\left|\sum_{n=1}^{N_v}v(n)e^{-j(2\pi/N_v)(k-1)(n-1)}\right|\right\} \\ &\qquad \times\cos\left(\frac{2\pi}{N_I}(i-1)(m-1)\right), \end{aligned} \tag{2.17}$$

donde N_I es la cantidad de bandas de integración o filtros, B_i son los límites de dichas bandas, ω_T es una ventana triangular y N_v es la longitud del tramo de voz.

2.4.3. Predicción lineal perceptual

En el modelo de predicción lineal perceptual (PLP, del inglés *perceptual linear prediction*), introducido por Hermansky [Hermansky, 1990], se modifica el espectro de la señal de voz para aproximar la respuesta del sistema auditivo humano. De esta manera se minimizan las diferencias entre hablantes y se reduce el efecto del ruido, pero preservando la información relevante para el reconocimiento, lo que lo hace más robusto en condiciones no ideales. Este espectro de potencia

perceptualmente motivado se obtiene mediante un banco de filtros similares a los descriptos en la Sección 2.4.2. Sin embargo, en este caso el escalado no lineal del espectro es realizado en base a las bandas críticas de la escala de *Bark*[1] [Huang et al., 2001].

Para obtener los coeficientes PLP se utiliza la recursión de Levinson-Durbin como en el caso de los coeficientes de predicción lineal [Makhoul, 1975b]. Sin embargo, los coeficientes de auto-correlación no se calculan en el dominio temporal sino que se obtienen a partir de la IFT del espectro de potencia de la señal, el cual es previamente alterado según la escala de Bark.

Posteriormente Hermansky y Morgan [Hermansky y Morgan, 1994] introdujeron una nueva representación basa en PLP denominada espectro relativo PLP (RASTA-PLP), en la cual se utiliza un banco de filtros pasa-banda para reducir el efecto de variaciones lentas del ruido ambiente.

2.4.4. Coeficientes delta y aceleración

El desempeño de los sistemas de reconocimiento de habla mejora considerablemente cuando se añaden las derivadas temporales a las representaciones estáticas básicas [Young et al., 2001]. Esta sencilla alternativa aplicada a los MFCC provee información sobre los cambios espectrales que ocurren entre segmentos consecutivos de la señal de voz, lo que resulta útil para discriminar fonemas no sonoros. Los coeficientes delta o derivadas de primer orden, correspondientes al t-ésimo tramo de una señal, se calculan a partir de los MFCC, $C_{mel}(m,t)$, mediante la siguiente regresión:

$$D(m,t) = \frac{\sum_{\delta=1}^{\Delta} \delta(C_{mel}(m,t+\delta) - C_{mel}(m,t-\delta))}{2\sum_{\delta=1}^{\Delta} \delta^2}, \tag{2.18}$$

donde Δ indica el tamaño de la ventana, es decir la cantidad de tramos anteriores y posteriores que se tienen en cuenta para el cálculo, que típicamente toma los valores 1 o 2. Con esta misma fórmula, pero aplicada sobre los coeficientes delta, se obtienen las derivadas de segundo orden, o coeficientes de aceleración [Haque et al., 2009].

[1] La frecuencia en Barks puede ser expresada en términos de la frecuencia lineal en Hertz mediante $F_{Bark}(F_{Hz}) = 13\arctan(0{,}0076\,F_{Hz}) + 3{,}5\arctan\left(F_{Hz}/7500\right)^2)$.

Capítulo 3

Coeficientes Cepstrales Evolutivos

3.1. Introducción

Como se mencionó en la Sección 2.4.2, la representación más utilizada actualmente en los sistemas de reconocimiento automático del habla es la de los MFCC. Sin embargo, debido a la degradación que sufre el desempeño de los reconocedores automáticos en presencia de ruido, se han estudiado diversas alternativas y se han logrado importantes avances en el desarrollo de nuevas técnicas robustas de extracción de características. Por ejemplo, recientemente, se introdujeron distintas modificaciones sobre ésta representación biológicamente inspirada [Bŏril et al., 2006; Nasersharif y Akbari, 2007; Wu y Cao, 2005; Zhou et al., 2007]. En este sentido, Slaney propuso un banco de filtros alternativo al utilizado comúnmente en la representación clásica [Slaney, 1998]. En este caso se obtuvieron algunas mejoras utilizando filtros de igual área. Otra novedosa alternativa es la que fue propuesta por Skowronski y Harris [Skowronski y Harris, 2002, 2003], que consiste en coeficientes cepstrales obtenidos mediante filtros de mayor ancho de banda, diseñados para aproximar las bandas críticas en el sistema auditivo humano. Con el uso de los denominados coeficientes cepstrales con factor humano (HFCC, del inglés *human factor cepstral coefficients*) los autores reportaron mejoras considerables sobre los tradicionales MFCC. Por otro lado, la ponderación de los MFCC de acuerdo a la relación señal-ruido en cada banda fue propuesta en [Yeganeh et al., 2008], y la utilización de la técnica de análisis discriminante lineal para optimizar un banco de filtros fue estudiada en [Burget y Hermansky, 2001]. Además, distintos enfoques de optimización empleando estrategias evolutivas fueron propuestos para la tarea de identificación del hablante [Charbuillet et al., 2007, 2009].

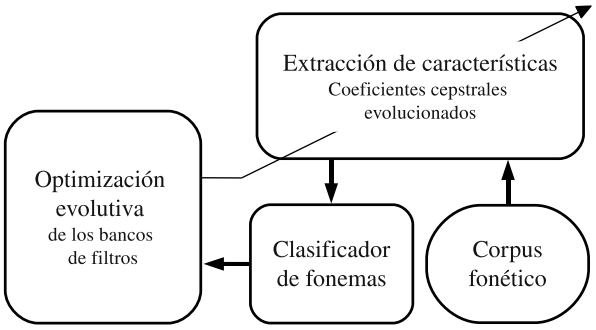


Figura 3.1. Esquema general de la propuesta de optimización de bancos de filtros.

Si bien los distintos enfoques mencionados han permitido mejorar la robustez de la representación tradicional bajo ciertas condiciones, surge el interrogante de si existe alguna alternativa realmente óptima para la tarea de reconocimiento.

En este capítulo se desarrollan dos propuestas para optimizar el banco de filtros utilizado en el cálculo de los coeficientes cepstrales, mediante técnicas de optimización evolutiva. En la Figura 3.1 se puede observar un esquema general válido para ambas propuestas. Cada individuo en la población del EA codifica un banco de filtros diferente, y para evaluar qué tan bueno resulta cada uno de ellos para parametrizar las señales de voz se utiliza un clasificador de fonemas basado en HMM. El objetivo del método propuesto es encontrar un banco de filtros más apropiado, en el sentido de que la representación resultante permita obtener mejores resultados de clasificación en comparación con los tradicionales MFCC. Algunos resultados preliminares obtenidos mediante esta propuesta fueron reportados en [Vignolo et al., 2009].

Este problema también podría abordarse mediante un MOEA [Coello Coello et al., 2007], para optimizar de manera conjunta la tasa de clasificación y otras características de la representación indirectamente relacionadas. Por ejemplo, podría tenerse en cuenta la dependencia estadística de los coeficientes de la representación, así como una medida de la gaussianidad de la distribución de los mismos. Sin embargo, teniendo en cuenta que el objetivo principal es mejorar la tasa de clasificación, mientras que las otras medidas no representan objetivos en sí mismas, y están subordinadas al objetivo principal, se aborda la optimización considerando este único objetivo.

Este capítulo se organiza de la siguiente manera. En primer lugar se presenta

un método para la selección adaptativa del conjunto de datos en la optimización, empleado en los experimentos para mejorar la capacidad de generalización sin incrementar el tiempo de la evolución. Luego se describen las dos propuestas de optimización mencionadas. En la primer estrategia los parámetros de los bancos de filtros se codifican de forma directa en el cromosoma. A diferencia de la propuesta de Charbuillet y otros [Charbuillet et al., 2009], en la que sólo se optimizaron las frecuencias de inicio y fin de un par de bancos de filtros, aquí se propone optimizar tres parámetros frecuenciales por cada filtro dentro del banco de filtros. Esto resulta en un problema de optimización complejo, debido al gran tamaño de los cromosomas. Luego, se introduce una alternativa en la cual se codifican los parámetros mediante funciones de tipo *spline*, para reducir el tamaño de los cromosomas y el espacio de búsqueda. Seguidamente se describen los dos corpus fonéticos utilizados en la experimentación. Por último, se analizan y comparan los resultados obtenidos para cada una de las alternativas.

3.2. Selección adaptativa del conjunto de datos

Un problema común en optimización evolutiva es que, dependiendo de la función objetivo, se requiere mucho tiempo de cómputo. En muchos casos la evaluación de los individuos es lo que insume la mayor cantidad de tiempo, y esto es debido a que requiere la ejecución de cierto proceso sobre un conjunto de datos específicos del problema. En el problema que se intenta resolver en esta tesis, por ejemplo, se requiere entrenar y evaluar un clasificador sobre un corpus de habla. Esto implica que la cantidad de tiempo que requiere la evolución es proporcional al tamaño del conjunto de datos utilizado en la función aptitud, así como al tamaño de la población y la cantidad de generaciones. Por otro lado, el tamaño del corpus de datos empleado en la evaluación influye dramáticamente en la capacidad de generalización de las soluciones potenciales. Existe, por lo tanto, un compromiso entre la capacidad de generalización de la mejor solución que se puede obtener con el EA y el tiempo que demanda la evolución. Una solución a este problema consiste en realizar la optimización sobre un conjunto de datos adaptativo, es decir, que es modificado durante la evolución. Más precisamente, en cada generación los conjuntos de entrenamiento y prueba se conforman con distintos datos. Una idea similar es la utilización de conjuntos de datos dinámicos introducida originalmente por Zhang y otros para realizar el entrenamiento de redes neuronales de manera eficiente [Zhang y Cho, 1999; Zhang y Veenker,

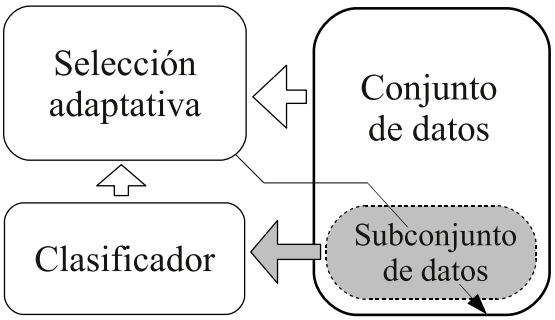


Figura 3.2. Esquema del método adaptativo de selección del conjunto de datos.

1991]. Motivados por estos trabajos, Gathercole y otros introdujeron métodos para la selección de subconjuntos de entrenamiento en programación genética [Gathercole y Ross, 1994]. Es así como estos principios podrían también ser de utilidad en estrategias optimización evolutiva, permitiendo mejorar la capacidad de generalización de las posibles soluciones sin incrementar el tamaño del corpus, y por lo tanto el costo computacional.

Mientras que en [Gathercole y Ross, 1994] se planteaba la adaptación de un único conjunto de datos, en este trabajo se propone realizar la adaptación tanto de un conjunto de entrenamiento como de un conjunto de prueba, en cada generación y durante toda la evolución. Para el caso del conjunto de prueba, la idea es dirigir la evolución hacia los casos que hayan causado más errores de clasificación en las generaciones previas, y también hacia aquellos que hayan sido incorporados en el subconjunto de prueba por un número de generaciones. Otra diferencia con el método propuesto en [Gathercole y Ross, 1994] es que el tamaño del subconjunto de entrenamiento, así como el de prueba, en este caso permanece constante durante toda la evolución.

En la Figura 3.2 se puede observar un esquema que ilustra este método de selección adaptativa del conjunto de datos [Vignolo et al., 2011a]. El algoritmo consiste en seleccionar aleatoriamente una cantidad determinada de patrones de entrenamiento y prueba en cada generación. Mientras que en el conjunto de entrenamiento todos los patrones tienen siempre la misma probabilidad de ser seleccionados en cada generación, a los patrones de prueba se les asigna una probabilidad que se calcula para generación. Para cada patrón de prueba dicha probabilidad se obtiene teniendo en cuenta la cantidad de veces que el mismo fue clasificado incorrectamente y la cantidad de generaciones que han transcurrido desde la últi-

ma vez en que éste fue incorporado en el subconjunto de prueba. Es decir, que en cada generación aquellos patrones de prueba que resultan más difíciles y aquellos más antiguos son los que tienen más chance de ser seleccionados.

En la primer generación, a todos los patrones de prueba se les asigna la misma probabilidad. Luego, se calcula un peso $W_k(g)$, para cada generación g y para cada patrón de prueba k como

$$W_k(g) = D_k(g)^d + A_k(g)^a, \tag{3.1}$$

donde $D_k(g)$ representa su dificultad actual y $A_k(g)$ la cantidad de generaciones desde la última en la que este patrón fue seleccionado. Luego la probabilidad de seleccionar el patrón de prueba k en la generación g está dada por:

$$P_k(g) = \frac{S\, W_k(g)}{\sum_j W_j(g)}, \tag{3.2}$$

donde S es el tamaño del subconjunto de prueba. En una generación g, para cada patrón k su antigüedad actual $A_k(g)$ se establece en 1 si este fue seleccionado y de lo contrario se incrementa en 1. Además, mientras se evalúa la población, la dificultad D_k, se incrementa en 1 cada vez que éste es clasificado incorrectamente.

3.3. Codificación directa

El MFB de la Figura 2.9, comúnmente utilizado para calcular los coeficientes cepstrales, revela que la búsqueda de un banco de filtros más apropiado puede basarse en el ajuste de varios parámetros, como la forma, la ganancia, la posición y el ancho de banda de cada filtro. Sin embargo, la optimización simultánea de todos estos parámetros resulta extremadamente compleja, razón por la cual se optó por mantener algunas de estas variables fijas.

En primera instancia se consideraron filtros triangulares, cada uno de ellos determinados por tres parámetros. Estos parámetros corresponden a la posición en frecuencia donde cada filtro comienza, donde alcanza su valor máximo de ganancia, y donde termina. Esto está esquematizado e la Figura 3.3, donde los tres parámetros mencionados se representan como a_i, b_i y c_i respectivamente. Dichos parámetros se codifican en los cromosomas como valores enteros, indexando las muestras en el dominio frecuencial. El ancho de banda y el solapamiento de los filtros se dejaron irrestrictos en este caso, y el número de filtros también fue optimizado, incorporando un gen adicional en los cromosomas (n_f en la Figura

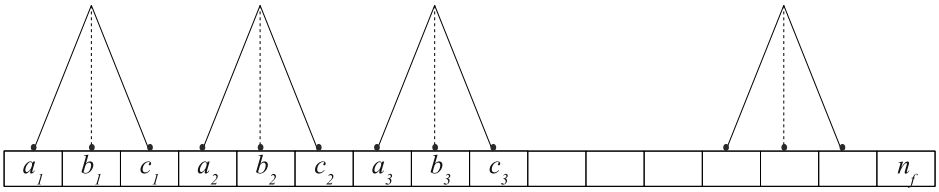


Figura 3.3. Esquema de la codificación directa de los parámetros en los cromosomas.

3.3). Este último elemento en un cromosoma indica que los primeros n_f filtros están activos. Luego, el largo de los cromosomas es tres veces la cantidad máxima de filtros, más uno.

En una segunda aproximación se optó por reducir la cantidad de parámetros en la optimización. En este caso, la optimización consistió en distribuir los filtros a lo largo de la banda frecuencial de interés, con restricción en el solapamiento. Esto significa que solamente las posiciones de máxima ganancia (parámetros b_i en Figura 3.3) del filtro fueron optimizadas, y el ancho de banda de cada filtro fue ajustado según el parámetro del filtro anterior y del filtro siguiente (por ejemplo, en el caso de la Figura 3.3, $c_2 \leftarrow b_3$ y $a_3 \leftarrow b_2$). De igual manera que en la alternativa anterior, en este caso la cantidad de filtros también fue optimizada.

Cada cromosoma representa un banco de filtros diferente, y cada uno de ellos se inicializa con un número aleatorio de filtros activos. Los cromosomas son codificados como cadenas de números enteros y el rango de valores posibles está determinado por la cantidad de muestras en el dominio frecuencial. La posición de los filtros también se inicializa de manera aleatoria según una distribución discreta uniforme sobre la banda frecuencial de 0 Hz hasta la mitad de la frecuencia de muestreo. Dicha posición determina la posición en frecuencia donde el filtro triangular alcanza su máxima ganancia, y para el caso de filtros de tres parámetros, las dos posiciones laterales de cada filtro se inicializan aleatoriamente mediante una distribución Binomial [Kay, 2006] centrada en la posición de máxima ganancia del mismo. Antes de aplicar los operadores de variación, los filtros en cada cromosoma son ordenados respecto a su posición en forma creciente.

Si bien el estado del arte establece que el método de selección por torneos permite acelerar la convergencia de un EA en general, en este caso se utiliza el método de la rueda de ruleta [Eiben y Smith, 2003], ya que las pruebas realizadas no mostraron ventajas de un método de selección sobre otro. Además, se incorpora la estrategia de elitismo en la búsqueda, la cual se ha probado beneficiosa para

la convergencia del algoritmo bajo ciertas condiciones [Bäck, 1996]. Se adoptó la estrategia de reemplazo generacional [Eiben y Smith, 2003] y los operadores de variación utilizados son el de mutación y el de cruza, implementados como se explica a continuación. La mutación de un filtro consiste en el desplazamiento aleatorio de uno de sus parámetros, y esta modificación es realizada mediante una distribución Binomial centrada en el valor actual de dicho parámetro. Este operador de mutación también puede cambiar, con la misma probabilidad, el número de filtros activos en el cromosoma. Por su parte, el operador de cruza (de un punto) intercambia filtros completos entre dos cromosomas padre. Para explicar su funcionamiento, supongamos que se aplica el operador de cruza sobre dos cromosomas padre, que denominaremos cromosoma A y cromosoma B. Luego, si B contiene más filtros activos que A, el punto de cruza será un número aleatorio entre 1 y el valor del parámetro n_f de A. Todos los genes más allá del punto de cruza en ambos cromosomas son intercambiados, resultando en un cromosoma hijo con la misma cantidad de filtros activos (n_f) que A y un cromosoma hijo con la misma cantidad de filtros activos que B.

La selección de los individuos para la reproducción se realiza según las bondades del banco de filtros representado por cada cromosoma. En este proceso se asigna mayor probabilidad a los cromosomas que proveen las mejores parametrizaciones, y éstos son aquellos que permitan obtener los mejores resultados de clasificación. Como se mencionó anteriormente, la función de aptitud propuesta consiste en un clasificador de fonemas, y la tasa de clasificación se asigna como valor de aptitud del individuo bajo evaluación. A los coeficientes obtenidos mediante esta estrategia evolutiva, para codificar las señales de voz, los llamaremos coeficientes cepstrales evolutivos (ECC, del inglés *evolutionary cepstral coefficients*).

3.4. Codificación mediante splines cúbicos

La optimización en simultáneo de varios parámetros de los bancos de filtros resulta en un problema complejo. Más aún, los resultados obtenidos en los experimentos de optimización de tres parámetros por cada filtro mostraron que se está tratando con un problema mal condicionado [Vignolo et al., 2009]. Por éste motivo, en la sección anterior se propusieron distintas alternativas considerando diferentes cantidades de parámetros libres. Para reducir el tamaño del cromosoma y el espacio de búsqueda, aquí se propone la codificación de los bancos de filtros mediante funciones de tipo spline. La elección de este tipo de funciones se debe

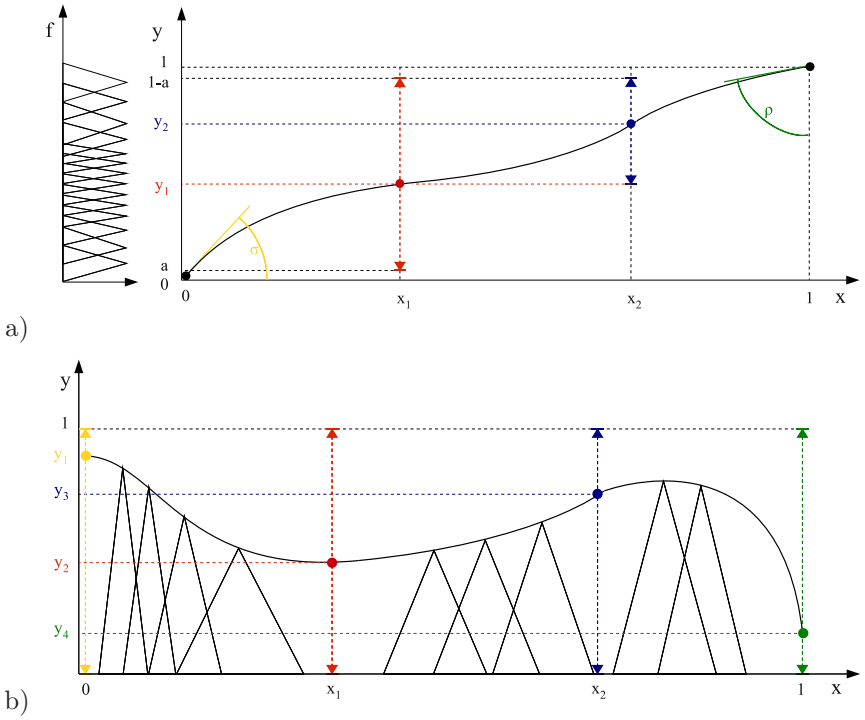


Figura 3.4. Esquemas que ilustran el uso de los splines en la optimización de bancos de filtros. a) Optimización de un spline para determinar la posición de los filtros; b) optimización de un spline para determinar la amplitud de los filtros.

a que facilitan la aplicación de restricciones necesarias en rango y dominio. Dichas restricciones resultan necesarias, por ejemplo, para que todos los bancos de filtros candidatos cubran todo el rango de frecuencias de interés y para favorecer la regularidad de los mismos [Vignolo et al., 2011b].

Se denota a la curva definida por un spline como $y = c(x)$, donde la variable x toma n_f valores equidistantes en el dominio $(0, 1)$, y a cada uno de éstos corresponde un valor en el rango $[0, 1]$. Aquí, n_f es el número de filtros en un banco de filtros, luego cada valor $x[i]$ está asociado a un filtro i, para $i = 1, ..., n_f$. De ésta manera, los valores en el rango de la función, mapeados desde 0 Hz hasta la mitad de la frecuencia de muestreo, determinan las posiciones en frecuencia donde cada uno de los filtros triangulares alcanza su máxima ganancia.

Se propone la adaptación de dos splines: el primero para ubicar en frecuencia una cantidad fija de filtros, y el segundo para determinar la amplitud máxima de cada uno de los filtros.

Splines para optimizar la distribución de los filtros: en este caso los splines son monótonamente crecientes y restringidos de manera que $c(0) = 1$ y $c(1) = 1$. Los parámetros libres son: dos valores y_1 y y_2 en el dominio del spline, correspondientes a dos valores fijos x_1 y x_2 en el rango del spline; además de los valores correspondientes a la derivada del spline en el punto $x = 0$, σ, y la derivada del spline en el punto $x = 1$, ρ. En la Figura 3.4(a) se puede observar un esquema que ilustra la relación entre un spline adaptado y un banco de filtros optimizado, y se indican los parámetros mencionados. Como se requiere que estos splines sean monótonamente crecientes, el valor del parámetro y_2 nunca puede ser menor que el valor del parámetro y_1. Luego, el parámetro y_2 se obtiene como $y_2 = y_1 + \delta_{y_2}$, y los parámetros que se codifican en los cromosomas son y_1, δ_{y_2}, σ y ρ. Dado un cromosoma particular, que asigna un conjunto determinado de valores a estos parámetros, los valores $y[i]$ correspondientes a los $x[i]$ son obtenidos mediante interpolación usando [Press et al., 1992b]

$$y[i] = P[i]y_1 + Q[i]y_2 + R[i]y_1'' + S[i]y_2'', \tag{3.3}$$

donde y_1'' y y_2'' son las segundas derivadas en los puntos x_1 y x_2 respectivamente. $P[i]$, $Q[i]$, $R[i]$ y $S[i]$ se definen como

$$P[i] \triangleq \frac{x_2 - x[i]}{x_2 - x_1}, \qquad R[i] \triangleq \frac{1}{6}((P[i])^3 - P[i])(x_2 - x_1)^2, \tag{3.4}$$

$$Q[i] \triangleq 1 - P[i], \qquad S[i] \triangleq \frac{1}{6}((Q[i])^3 - Q[i])(x_2 - x_1)^2. \tag{3.5}$$

Sin embargo, para calcular los valores $y[i]$ interpolados usando (3.3) se requiere conocer las segundas derivadas y_1'' y y_2'', que en general se desconocen. En el caso de los splines cúbicos se supone que la primer derivada es continua entre cada par de intervalos consecutivos, y esta restricción permite obtener ecuaciones para las segundas derivadas y_l'' [Press et al., 1992b]. Dichas ecuaciones se obtienen igualando la derivada de (3.3) evaluada para x_j en el intervalo (x_{j-1}, x_j) a la derivada de (3.3) evaluada para x_j en el intervalo (x_j, x_{j+1}). De esta manera se obtiene un sistema de ecuaciones en el que, para obtener una única solución, es necesario fijar condiciones de borde en $x = 0$ y $x = 1$. Para ésto se pueden fijar los valores de y en $x = 0$ y $x = 1$, o sus derivadas σ y ρ.

Los valores $y[i]$ son luego mapeados linealmente a valores en el rango de frecuencias de interés (desde 0 Hz hasta la mitad de la frecuencia de muestreo)

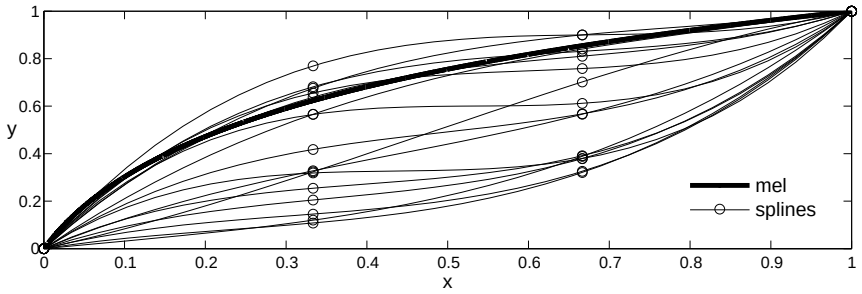


Figura 3.5. Comparación entre la escala de mel y el mapeo proporcionado por algunos splines.

para determinar la frecuencia donde cada uno de los n_f filtros alcanza su valor máximo, f_i^c, mediante:

$$f_i^c = \frac{(y[i] - y_{min}) f_s}{y_{max} - y_{min}}, \tag{3.6}$$

donde y_{min} y y_{max} son los valores mínimo y máximo del spline, respectivamente. Como se puede observar en la Figura 3.4(a), para los segmentos donde y asciende rápido los filtros se ubican relativamente lejos entre sí, mientras que donde y crece lentamente los filtros quedan posicionados cerca unos de otros. El parámetro `a` en la Figura 3.4(a) controla el rango de variación de y_1 y δ_{y_2} (y y_2), y es establecido de manera de reducir el número de splines con valores fuera de $[0, 1]$. Los cromosomas que producen splines cuyos valores escapen de estos límites son penalizados, y además al momento de evaluarlos la curva correspondiente se modifica de manera que $0 \leq y[i] \leq 1$ para todo i (esta modificación no se refleja en el cromosoma). La Figura 3.5 muestra algunos ejemplos de splines que cumplen las restricciones, y son comparados con el mapeo clásico de la escala de mel.

Splines para optimizar la amplitud de los filtros: aquí la única restricción es que y varíe dentro del rango $[0, 1]$, y los valores del spline para $x = 0$ y $x = 1$ no estén fijos. Luego, en este caso, los parámetros de la optimización son los valores y_1, y_2, y_3 y y_4, correspondientes a las constantes x_1, x_2, x_3 y x_4, respectivamente. Los valores $y[i]$ interpolados con estos splines determinan la amplitud máxima de cada uno de los n_f filtros. Esto está representado en la Figura 3.4(b), donde la ganancia de cada filtro está ponderada de acuerdo a un spline. Por lo tanto, se espera que las bandas frecuenciales relevantes para la clasificación sean destacadas, mientras que se reste importancia a aquellas que

Algoritmo 2: Optimización para obtener los CCES.

Inicializar la población del EA
Inicializar $P_k(g) = 1$ para todo k
Elegir subconjuntos y actualizar $A_k(g)$
Evaluar población
Actualizar $D_k(g)$ en base a los resultados de clasificación
repetir
  Seleccionar padres (método ruleta)
  Crear la nueva población
  Reemplazar la población actual
  Dados $A_k(g)$ y $D_k(g)$ obtener $P_k(g)$ usando (3.2) y (3.1)
  Elegir subconjuntos y actualizar $A_k(g)$
  Evaluar población
  Actualizar $D_k(g)$ en base a los resultados de clasificación
hasta *alcanzar el criterio de finalización*

estén más contaminadas con ruido.

Debe notarse que, como será explicado en la Sección 3.4, utilizando esta codificación el tamaño del cromosoma se reduce de n_f a 4. Por ejemplo, para un banco de filtros típico de 30 filtros, el tamaño del cromosoma se reduce de 30 a 4. Además, para el esquema de optimización completa en el cual se optimizan también las amplitudes de los filtros, el tamaño del cromosoma se reduce de 60 a 8 genes. Esto es gracias a que, con la codificación propuesta, el tamaño del cromosoma es independiente de la cantidad de filtros.

Descripción del proceso de optimización

En la población del EA cada individuo codifica los parámetros de los splines para generar distintos bancos de filtros, dando lugar a fórmulas alternativas para los coeficientes cepstrales evolucionados mediante splines (CCES). Un cromosoma se codifica como una cadena de números reales, inicializados aleatoriamente por medio de una distribución de probabilidad uniforme. El tamaño de los cromosomas está dado por el número de splines adaptados multiplicado por el número de parámetros libres en cada spline. Se propone, en primer lugar, la adaptación de un único spline para optimizar la posición de cada filtro; y en segundo lugar la adaptación de dos splines en simultáneo para optimizar la posición y la amplitud

Algoritmo 3: Evaluar población.

para *cada individuo en la población*
- Obtener los valores $y[i]$ del 1º spline (3.3) a partir de y_1, y_2, σ y ρ
- Dado $y[i]$, obtener las posiciones frecuenciales de los filtros f_i^c usando (3.6)
- Obtener los valores $y[i]$ del 2º spline (3.3) a partir de y_1, y_2, y_3 y y_4
- Ajustar la amplitud de cada filtro $y[i]$
- Construir los filtros triangulares ω_T (2.17)
- Calcular los CCES usando (2.17)
- Entrenar el clasificador basado en HMM con el subconjunto de entrenamiento elegido
- Probar el clasificador con el subconjunto de prueba elegido
- Asignar la tasa de clasificación como aptitud del individuo evaluado

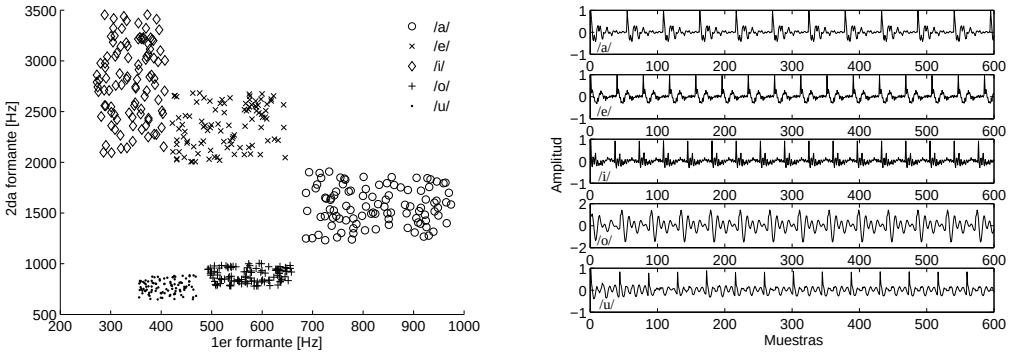
de cada filtro. De esta manera, para estos dos casos los cromosomas son apenas de tamaño 4 y 8, respectivamente.

Al igual que en la estrategia evolutiva presentada anteriormente, en este caso se utiliza el método de selección basado en la rueda de ruleta y elitismo. Se emplea un operador de mutación que consiste en la modificación aleatoria de los parámetros de los splines, a partir de una distribución uniforme. Por su parte, el operador de cruza clásico, en este caso intercambia los parámetros de los spline entre un par de cromosomas.

Los pasos para el proceso de optimización de bancos de filtros con codificación mediante splines están resumidos en el Algoritmo 2. Los detalles del procedimiento para evaluar la población se muestran en el Algoritmo 3.

3.5. Descripción del corpus de habla

En esta sección se describen los corpus fonéticos utilizados en la experimentación. En primer lugar se dan los detalles del corpus sintetizado que se utilizó para ajustar adecuadamente los parámetros del EA, manteniendo las condiciones experimentales controladas. Luego se detalla el corpus de fonemas reales con el que se realizó la mayor parte de la experimentación.



(a) (b)

Figura 3.6. Corpus fonético sintetizado. a) Distribución de las primeras dos formantes. b) Ejemplos de cada vocal.

3.5.1. Fonemas del español sintetizados

El corpus fonético sintetizado incluye las vocales del español (fonemas `/a/`, `/e/`, `/i/`, `/o/` y `/u/`), ya que estos fonemas pueden ser simulados de manera controlada. Estos fonemas fueron modelados a partir de los coeficientes de predicción lineal [Rabiner y Juang, 1993] obtenidos de pronunciaciones reales. Cada ocurrencia fue sintetizada con una F_0 aleatoria distribuida uniformemente en el rango de 80 a 250 Hz, simulando hablantes masculinos y femeninos. Para introducir una variabilidad controlada, las formantes F_1 y F_2 también fueron modificadas aleatoriamente. De esta manera se simularon 1300 ocurrencias para cada fonema, es decir, 6500 señales en total. La Figura 3.6 muestra la distribución de las formantes de los fonemas sintetizados y una realización de cada uno de ellos.

Estos fonemas fueron generados en forma aislada, simulando vocales sostenidas de 150 milisegundos de duración. Por cada segundo de señal se tienen 8000 muestras y se extrajeron tramos de 50 milisegundos (400 muestras) de cada fonema. Las señales de este corpus fueron contaminadas con ruido blanco aditivo de energía variable, de manera que cada señal tiene una relación señal-ruido (SNR, del inglés *Signal-to-noise ratio*) entre 2 y 10 dB.

3.5.2. Fonemas reales del inglés

El corpus de fonemas reales del inglés fue extraído de la base de datos TIMIT [Garofalo et al., 1993], la cual es la más difundida en el área. Las señales de voz

fueron elegidas aleatoriamente de entre todas las regiones, incluyendo hablantes masculinos y femeninos. Las elocuciones fueron segmentadas fonéticamente según las transcripciones fonéticas provistas en la base de datos. De esta manera se generaron archivos individuales con la señal temporal de cada una de las ocurrencias de los fonemas elegidos, que fueron luego contaminadas con ruido blanco aditivo en distintos niveles de SNR. Estas señales fueron muestreadas a una frecuencia de 16 kHz, y se extrajeron tramos de 25 milisegundos (400 muestras) con un paso de 12,5 milisegundos. De cada ocurrencia de los fonemas elegidos se extrajeron todos los tramos posibles, completando con ceros en los casos que fuese necesario. En el corpus utilizado en los experimentos de esta tesis se incluyeron los fonemas `/b/`, `/d/`, `/eh/`, `/ih/` y `/jh/`. Las consonantes oclusivas `/b/` y `/d/` se consideraron por ser éstas usualmente difíciles de distinguir en muchos contextos. El fonema `/jh/` para incluir las características especiales de los sonidos fricativos, y las vocales `/eh/` y `/ih/` porque se encuentran cercanas en el espacio de las formantes. Este grupo de fonemas usualmente se tiene en cuenta porque constituye un conjunto de clases que resulta difícil de clasificar [Stevens, 2000].

3.6. Resultados y discusión

3.6.1. Codificación directa

Fonemas del español sintetizados

Se realizaron diferentes corridas de la optimización, teniendo en cuenta distintas combinaciones de los parámetros del AE y de los parámetros relacionados a los bancos de filtros. Los mejores resultados fueron obtenidos optimizando solamente las posiciones de los filtros y la cantidad de los mismos, la cual variaba entre 17 y 32. En el EA se utilizó una población de 100 individuos, una tasa de cruza de 0,8, y una tasa de mutación de 0,1 para cada filtro en el cromosoma. El clasificador de fonemas, empleado como función de aptitud, utiliza HMM continuos de tres estados y las densidades de probabilidad de observación se modelan con mezclas de Gaussianas [Demuynck et al., 1998]. Se utilizaron herramientas del *HMM Toolkit* (HTK) [Young et al., 2000] para construir y manipular los HMM, para entrenar los parámetros mediante el algoritmo de Baum-Welch [Jelinek, 1999], y para buscar la secuencia de estados de máxima probabilidad mediante el algoritmo de Viterbi [Huang et al., 1990].

Durante las optimizaciones se emplearon 500 señales de entrenamiento y un

conjunto de 500 señales diferente para la prueba, para calcular la aptitud de cada individuo. En la elección del tamaño de estos conjuntos se tuvo en cuenta, además de los criterios relacionados al desempeño del clasificador, el costo computacional de la optimización. En este caso, los conjuntos de entrenamiento y prueba no sufrieron modificaciones durante la evolución, es decir, no se utilizó el algoritmo de selección adaptativa. Como criterio de finalización de la optimización, las evoluciones fueron terminadas luego de que transcurrieran 100 generaciones sin mejoras en la aptitud máxima de la población. Finalizada una evolución, se tomaron los mejores veinte bancos de filtros según su aptitud, y con ellos se realizaron pruebas de validación con un conjunto diferente de 500 señales. A partir de los resultados de estas pruebas se seleccionaron los dos mejores bancos de filtros, descartando aquellos que fueron sobre-optimizados.

La Tabla 3.1 resume los resultados de validación para los bancos de filtros obtenidos en dos optimizaciones diferentes, e incluye también los resultados de clasificación para el MFB estándar sobre los mismos conjuntos de datos. La cuarta columna de la tabla contiene los resultados de clasificación obtenidos empleando mezclas de Gaussianas con matrices de covarianza diagonal (MCD) para modelar las densidades de observación en los HMM, y la quinta columna contiene los resultados obtenidos considerando matrices de covarianza completa (MCC).

Los bancos de filtros evolucionados (BFE) 1 y 2 fueron obtenidos usando HMM con MCD en la función de aptitud durante la optimización, mientras que los BFE 3 y 4 fueron obtenidos usando HMM con MCC. Como puede observar, se obtuvieron bancos de filtros que se desempeñan mejor que el MFB cuando se utiliza HMM-MCC. También es importante notar que con MFB también se obtiene mejor resultado cuando se utilizan los HMM-MCC que cuando se emplea HMM-MCD.

La Figura 3.7 muestra los cuatro BFE mencionados en el párrafo anterior. Una característica en común que se puede observar es que presentan una alta densidad de filtros desde aproximadamente 500 a 1000 Hz, lo cual puede estar relacionado a la distribución de la primera formante en las vocales (Figura 3.6). Más aún, considerando la distribución de la segunda formante, se puede notar que estas agrupaciones de filtros permiten distinguir mejor los fonemas `/o/` y `/u/` de los otros. Otra característica común en estos cuatro BFE es que el rango de frecuencias de 0 a 500 Hz está cubierto sólo por dos filtros, a excepción del BFE 3, que tiene un además filtro de 0 a 40 Hz. Este filtro angosto podría aislar los picos en frecuencia cero de la información relevante. También tienen en común la característica de que, desde 1000 hasta 2500 Hz, los cuatro BFE muestran una

Banco de filtros	Número de filtros	Número de coeficientes	Test de Validación MCD	MCC
BFE 1	17	9	**95,20**	**97,00**
BFE 2	18	10	**95,40**	**96,80**
BFE 3	18	10	93,00	96,40
BFE 4	17	9	94,60	96,20
MFB	23	13	94,80	96,20
MFB	17	9	93,00	95,20

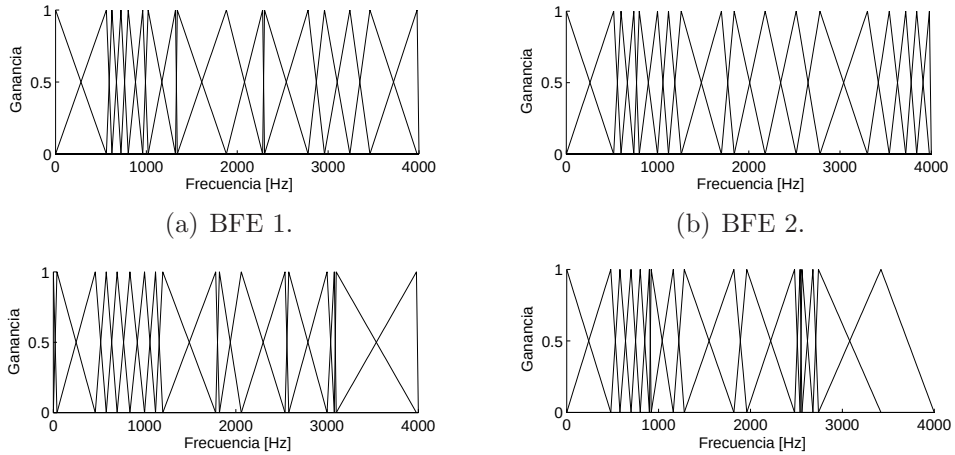
Tabla 3.1. Porcentajes de acierto promedio obtenidos con fonemas del español sintetizados.

distribución de filtros similar. Por otro lado, una particularidad del segundo BFE es que presenta alta resolución en altas frecuencias, en oposición al MFB, lo que le permite preservar información de otras formantes.

Fonemas del inglés

En este segundo grupo de experimentos, los mejores resultados fueron obtenidos en el caso de tres parámetros por cada filtro. En este caso el número de filtros en los cromosomas también podía variar entre 17 y 32. Para evaluar la aptitud de los individuos se utilizó una partición de datos adaptativa de 1000 señales de entrenamiento y 400 señales de prueba, y un clasificador basado en HMM con MCC. Al igual que en el caso anterior, el tamaño de estos conjuntos de datos se determinó de manera experimental y teniendo en cuenta el costo computacional de la optimización. La partición de datos utilizada durante la evolución fue remuestreada en cada generación de acuerdo a la estrategia descripta en la Sección 3.2, y las ocurrencias de fonemas fueron seleccionadas dinámicamente (en cada generación) de un total de 6045 señales disponibles para entrenamiento y 1860 señales disponibles para prueba. En base a la bibliografía [Gathercole y Ross, 1994] y a pruebas realizadas, los exponentes de dificultad y antigüedad fueron ambos establecidos en 1,0.

Es importante mencionar que, previo a los experimentos cuyos resultados se presentan en esta sección, fueron realizadas diferentes pruebas con este mismo grupo de fonemas. Entre las alternativas exploradas se han utilizado distintas configuraciones en la optimización, y diferentes parámetros libres en los bancos de filtros. Por otro lado, también se realizaron experimentos con las mismas con-



(a) BFE 1. (b) BFE 2.

(c) BFE 3. (d) BFE 4.

Figura 3.7. Bancos de filtros optimizados para los fonemas `/a/, /e/, /i/, /o/` y `/u/` del corpus sintetizado.

figuraciones a los aquí descriptos, en los cuales el conjunto de datos permanecía fijo durante la evolución. Los resultados obtenidos en dichos experimentos preliminares fueron publicados en [Vignolo et al., 2009]. Sin embargo, al realizar selección adaptativa del conjunto de datos se obtuvieron resultados de validación considerablemente mejores, haciendo evidente la importancia de esta estrategia.

Al igual que en el caso de los experimentos con fonemas sintetizados, las evoluciones fueron terminadas luego de que transcurrieran 100 generaciones sin mejoras en la aptitud máxima de la población. Para los parámetros del EA, en este caso, también se utilizó la configuración detallada para el caso de los fonemas sintetizados. Se realizaron pruebas de validación considerando diferentes niveles de ruido, y para cada nivel de ruido se emplearon diez particiones de datos diferentes, cada una de las cuales consistía en 2500 patrones de entrenamiento y 500 patrones de prueba (en este caso, para la elección del tamaño de estos conjuntos, ya no se tiene en cuenta el costo computacional).

Aquí se muestran los resultados de clasificación logrados con los BFEs obtenidos en tres evoluciones diferentes, que difieren únicamente en el nivel de ruido presente en las señales empleadas en la evaluación de los individuos. La Tabla 3.2 muestra los resultados promedio de clasificación obtenidos en la validación, comparando los bancos de filtros optimizados para señales con 0 dB SNR y el clásico MFB, usando HMM-MCD. Se realizaron las validaciones para los diez mejores

Banco de filtros	Número de filtros	Número de coeficientes	−5 dB	0 dB	20 dB	Limpio	Diferencia
BFE-A0	32	17	24,76	32,62	58,26	65,54	0,44
BFE-A1	17	9	20,26	26,02	62,16	62,62	−9,68
BFE-A2	21	11	20,16	21,34	59,56	60,00	−19,68
BFE-A3	29	15	24,34	32,92	66,08	64,32	6,92
BFE-A4	19	10	20,38	26,32	63,64	61,22	−9,18
BFE-A5	19	10	20,52	26,24	60,62	60,26	−13,10
BFE-A6	21	11	**31,10**	**35,78**	61,52	60,80	**8,46**
BFE-A7	29	15	22,58	30,52	63,90	64,58	0,84
BFE-A8	25	13	22,94	30,76	62,10	62,08	−2,86
BFE-A9	22	12	23,60	31,54	63,54	66,14	4,08
MFB	23	13	20,00	23,18	**68,40**	**69,16**	

Tabla 3.2. Porcentajes de clasificación promedio de diez particiones de datos con fonemas reales de inglés. Bancos de filtros optimizados para 0 dB SNR.

BFEs obtenidos en la optimización, y las distintas evaluaciones para cada uno de ellos se repitieron con cada una de las diez particiones de datos. Para la prueba del clasificador se consideraron diferentes SNR, pero entrenando con señales limpias en todos los casos (MMTT, del inglés *MisMatch Training and Test*). De esta manera el clasificador se evalúa en condiciones más cercanas a la realidad. Para el banco de filtros de referencia, MFB, se repiten las pruebas con las mismas condiciones y particiones de datos. Cada uno de los resultados de la tabla fue obtenido como promedio de los resultados de clasificación con diez particiones de datos. La última columna muestra la diferencia acumulada entre cada una de las diez primeras filas con la última, los valores más altos indican los mejores bancos de filtros. Por ejemplo, en la Tabla 3.2, se obtiene el valor 0,44 en la primera fila sumando la diferencia de los valores desde la cuarta columna hasta la séptima columna en la misma fila, con los respectivos de la última fila. Como el número de filtros es uno de los parámetros optimizados, se comparan todos los BFE con un MFB de 23 filtros, que es una configuración estándar en reconocimiento de habla. Se puede observar que en las pruebas de −5 y 0 dB SNR el BFE A6 se desempeña mucho mejor que el MFB. A partir de esto se puede decir que la distribución de los filtros en BFE A6 permite distinguir mejor las frecuencias formantes de las componentes frecuenciales del ruido. Esto significa que mediante el uso del banco de filtros optimizado se obtiene una representación más robusta que la parametrización estándar. La misma comparación se puede observar en las

Banco de filtros	Número de filtros	Número de coeficientes	−5 dB	0 dB	20 dB	Limpio	Diferencia
BFE-B0	20	11	20,04	22,24	62,30	63,06	−13,10
BFE-B1	19	10	22,18	30,06	53,76	64,12	−10,62
BFE-B2	22	12	22,44	30,24	60,68	64,96	−2,42
BFE-B3	19	10	21,38	27,84	68,08	67,80	4,36
BFE-B4	19	10	21,10	26,72	62,40	64,52	−6,00
BFE-B5	19	10	22,06	**34,54**	55,56	64,46	−4,12
BFE-B6	18	10	20,22	31,92	**68,44**	66,64	**6,48**
BFE-B7	19	10	**22,88**	31,98	64,44	67,26	5,82
BFE-B8	18	10	21,58	27,90	64,04	61,88	−5,34
BFE-B9	19	10	22,82	31,08	64,28	68,04	5,48
MFB	23	13	20,00	23,18	68,40	**69,16**	

Tabla 3.3. Porcentajes de clasificación promedio de diez particiones de datos con fonemas reales de inglés. Bancos de filtros optimizados para 20 dB SNR.

Tablas 3.3 y 3.4 para BFE optimizados usando señales con ruido a 20 dB SNR y señales limpias, respectivamente. Nuevamente, se observa que algunos BFEs se desempeñan considerablemente mejor que el MFB cuando las señales de prueba contienen altas cantidades de ruido, e incluso también hay mejoras en el caso de 20 dB SNR.

Como se puede observar, distintos BFE obtienen mejoras para distintas SNR. Sin embargo, esto no representa una desventaja importante, ya que en un sistema de RAH se podría estimar la cantidad de ruido presente en la señal para utilizar el BFE más conveniente.

A partir de estos tres grupos de BFE se seleccionaron los mejores y con éstos se hicieron pruebas de validación adicionales con ruido a 5, 10, 15 y 30 dB SNR. Los resultados, obtenidos como promedio de los porcentajes de acierto en diez particiones diferentes, se pueden apreciar en la Tabla 3.5, así como también los resultados para los bancos de filtros de referencia.

Para el caso de HFCC se consideraron 30 filtros, uno más que el número propuesto en [Skowronski y Harris, 2004] porque la frecuencia de muestreo de las señales consideradas en estos experimentos es mayor. El ancho de banda de los filtros en HFCC está controlado por un parámetro que los autores llamaron *E-factor*, que aquí fue establecido en 5, basando esta elección en los resultados de reconocimiento reportados en [Skowronski y Harris, 2004]. Como se sugiere en el trabajo mencionado, se consideraron los primeros 13 coeficientes cepstrales.

Banco de filtros	Número de filtros	Número de coeficientes	−5 dB	0 dB	20 dB	Limpio	Diferencia
BFE-C0	21	11	20,56	27,94	64,14	63,48	−4,62
BFE-C1	18	10	20,08	34,20	61,26	60,66	−4,54
BFE-C2	19	10	20,28	27,74	62,62	60,72	−9,38
BFE-C3	18	10	21,94	30,32	62,70	64,36	−1,42
BFE-C4	18	10	20,56	**36,88**	**69,82**	68,08	**14,60**
BFE-C5	18	10	22,26	30,42	65,14	63,40	0,48
BFE-C6	19	10	20,30	30,16	64,82	62,62	−2,84
BFE-C7	18	10	20,16	30,66	63,22	61,96	−4,74
BFE-C8	18	10	**26,52**	33,56	56,62	64,00	−0,04
BFE-C9	18	10	20,40	26,68	66,88	66,22	−0,56
MFB	23	13	20,00	23,18	68,40	**69,16**	

Tabla 3.4. Porcentajes de clasificación promedio de diez particiones de datos con fonemas reales de inglés. Bancos de filtros optimizados para señales limpias.

El banco de filtros de Slaney consistía en 40 filtros, como se propuso en [Slaney, 1998], y se calcularon los 20 primeros coeficientes cepstrales. Se puede ver que con los BFEs se obtiene un mejor desempeño del clasificador, comparado con el clásico MFB, cuando la cantidad de ruido en las señales de prueba es mayor que la cantidad de ruido en las señales de entrenamiento. Más aún, con BFE C4 y BFE B6 se obtienen mejores resultados que con el banco de filtros de Slaney para todos las cantidades de ruido, excepto en −5 dB SNR. Por otro lado, los BFEs permiten lograr tasas de acierto más altas que el HFCC para las SNR más bajos, es decir, desde −5 dB hasta 15 dB SNR. Las mejoras obtenidas se pueden apreciar mejor en la Figura 3.8, donde se puede observar que los resultados del BFE C4 superan a los del MFB en el rango de 0 dB hasta 15 dB SNR.

Si bien los resultados del MFB no son superados para los casos de 30 dB SNR y señales limpias, este comportamiento es común en la mayoría de las parametrizaciones robustas [Gong, 1995]. Por ejemplo, el banco de filtros HFCC también permite obtener mejores resultados que el MFB en las condiciones de ruido más desfavorables, sin embargo, por encima de 20 dB SNR las mejoras no son considerables. Además, como es habitual, en este caso también ocurre que la degradación en el desempeño de reconocimiento es proporcional a la diferencia entre la cantidad de ruido presente en las señales de prueba y la cantidad de ruido en las señales de entrenamiento [Davis, 2002; Zhou et al., 2007].

La Figura 3.9 muestra algunos BFEs seleccionados de la Tabla 3.5. Como se

BF	−5 dB	0 dB	5 dB	10 dB	15 dB	20dB	30 dB	Limpio
BFE-A3	24,34	32,92	37,68	46,36	52,98	66,08	65,04	64,32
BFE-A6	**31**,**10**	35,78	44,38	46,88	53,12	61,52	60,36	60,80
BFE-B6	20,22	31,92	55,12	67,20	**68**,**84**	68,44	67,20	66,64
BFE-B7	22,88	31,98	36,86	44,42	49,64	64,44	67,58	67,26
BFE-C4	20,56	**36**,**88**	**60**,**30**	**68**,**32**	68,70	69,82	67,42	68,08
BFE-C5	22,26	30,42	34,38	44,32	57,28	65,14	63,52	63,40
MFB	20,00	23,18	37,90	44,68	51,42	68,40	**69**,**80**	69,16
HFCC	20,24	25,98	47,26	62,78	67,68	**70**,**54**	69,42	**70**,**36**
Slaney	29,94	30,28	36,44	54,76	60,66	62,02	61,52	62,78

Tabla 3.5. Porcentajes de clasificación obtenidos con fonemas del inglés. Promedio sobre diez particiones de datos

mencionó anteriormente, una característica que todos tienen en común es el amplio ancho de banda de los filtros, comparado con los del MFB. Esto coincide con el estudio realizado en [Skowronski y Harris, 2004] sobre el efecto de la ampliación de los filtros en la robustez de la representación. En todos los BFEs podemos notar también que existe mucho solapamiento entre los filtros, ya que no había ninguna restricción sobre ésto en la optimización. Sin embargo, el aumento del solapamiento resulta en coeficientes cepstrales más correlacionados, por lo cual puede resultar conveniente emplear HMM con matrices de covarianza completas. Se puede observar un agrupamiento de un número relativamente alto de filtros en la banda de frecuencias de 0 a 4000 Hz en el caso de BFE C4, el cual permite obtener los mejores resultados con señales de prueba ruidosas.

Para poder analizar la información que capturan estas representaciones, se realizó la reconstrucción de un estimado del espectro de magnitud de tiempo corto de las señales de voz, mediante el método propuesto en [Ellis, 2005]. Este método consiste en escalar el espectrograma de una señal de ruido blanco mediante el espectro de magnitud reconstruido a partir de los coeficientes cepstrales. Las Figuras 3.10 y 3.11 muestran los espectrogramas de la frase `SI648` del corpus TIMIT, con ruido blanco aditivo a 50 dB y 10 dB SNR respectivamente. La Figura 3.10 muestra que los filtros amplios del BFE difuminan los coeficientes de energía a lo largo del eje de frecuencias, por lo cual resulta más difícil distinguir las frecuencias formantes. Sin embargo, los resultados muestran que esa información no se pierde. Más aún, la clasificación de fonemas resulta más sencilla porque se elimina información relacionada a la frecuencia fundamental. Por otro lado, en

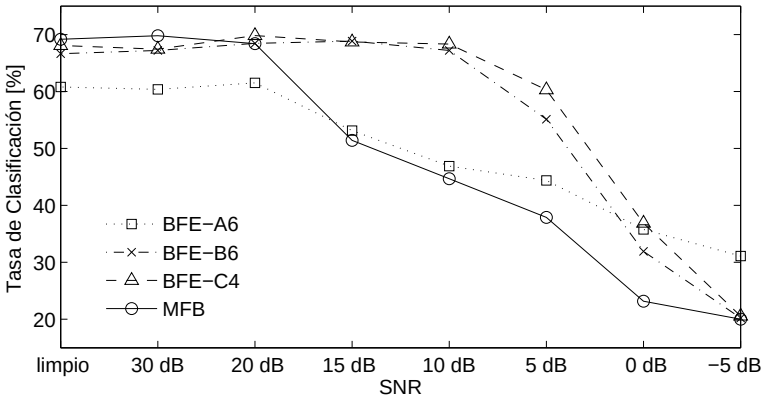
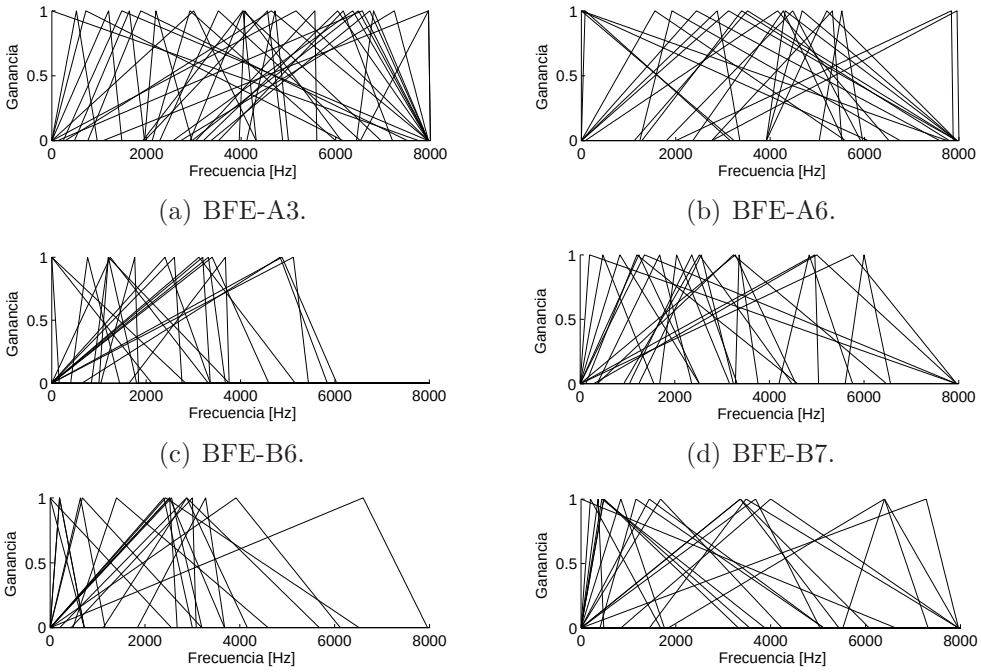


Figura 3.8. Desempeño obtenido con los mejores BFE comparado con el MFB (fonemas del inglés).

la Figura 3.11 se puede apreciar que, cuando la señal es ruidosa, la información relevante es más clara en el espectrograma reconstruido a partir de los CCE. Esto es debido a que la distribución de los filtros y los anchos de banda en el BFE C4 permiten conservar la información relevante de altas frecuencias, lo que no sucede en cuando se utilizan los MFCC.

La Tabla 3.6 exhibe las matrices de confusión para el MFB y el BFE C4, a partir de los resultados obtenidos con señales de prueba con ruido a 10 y 15 dB SNR. A partir de éstas matrices, se puede observar que los fonemas `/eh/` e `/ih/` son clasificados incorrectamente con MFB, mientras que con el BFE C4 se obtienen buenos resultados. De hecho, cuando la cantidad de ruido no es considerable, el desempeño de clasificación para los cinco fonemas es similar con ambos bancos de filtros. Sin embargo, a medida que disminuye la SNR, más falla el MFB en clasificar los fonemas `/eh/` e `/ih/`. Estos son confundidos mayormente con los fonemas `/b/` y `/d/`, mientras que la tasa de acierto para los otros fonemas apenas se ve afectada. Por otro lado, en el caso del BFE C4, el ruido degrada la tasa de acierto de manera uniforme para todos los fonemas, pero ninguno de ellos se ve tan afectado como en el caso del MFB. Esto significa que no solamente la tasa de acierto promedio es mayor, sino que también la varianza entre la tasa para los fonemas individuales es menor. A partir de estos resultados se puede concluir que el BFE provee una representación más robusta, que permite obtener mejor desempeño de reconocimiento en presencia de ruido [Vignolo et al., 2011a].



(a) BFE-A3. (b) BFE-A6.

(c) BFE-B6. (d) BFE-B7.

(e) BFE-C4. (f) BFE-C5.

Figura 3.9. Bancos de filtros optimizados para los fonemas `/b/`, `/d/`, `/eh/`, `/ih/` y `/jh/` del corpus TIMIT.

Dependencia estadística de los CCE

Usualmente, los coeficientes de energía obtenidos en la integración por bandas del espectro de magnitud son transformados mediante la CT hacia el dominio cepstral. Además de los aspectos teóricos mencionados en la Sección 2.4.2, ésto tiene los efectos de reducir la correlación entre los coeficientes y las dimensiones del vector de características. Si bien la CT discreta no puede decorrelacionar los coeficientes como ciertas transformaciones basadas en los datos [Wang et al., 2005], en la práctica se verifica que los MFCC están muy poco correlacionados [Kwon y Lee, 2004]. Esto es deseable para los reconocedores basados en HMM, en los cuales se modelan las densidades de observación mediante mezclas de Gaussianas con matrices de covarianza diagonal [Demuynck et al., 1998].

Sin embargo, esta suposición de una débil dependencia estadística entre los coeficientes no es cierta para los CCE. Como muestra la Figura 3.9, el ancho de

		MFB					BFE-C4				
		/b/	/d/	/eh/	/ih/	/jh/	/b/	/d/	/eh/	/ih/	/jh/
15 dB	/b/	**64**,**7**	34,8	00,0	00,0	00,5	**56**,**9**	39,7	01,8	01,4	00,2
	/d/	11,7	**83**,**2**	00,0	00,1	5,00	14,1	**79**,**9**	00,6	00,9	04,5
	/eh/	33,1	51,0	**05**,**0**	07,1	03,8	03,9	04,5	**73**,**5**	18,1	00,0
	/ih/	21,8	45,3	04,7	**18**,**9**	09,3	12,6	09,9	18,2	**59**,**3**	00,0
	/jh/	00,1	14,6	00,0	00,0	**85**,**3**	00,3	25,3	00,2	00,3	**73**,**9**
		Promedio: 51,42					Promedio: **68**,**70**				
10 dB	/b/	**55**,**4**	44,0	00,0	00,0	00,6	**48**,**8**	48,6	01,5	00,5	00,6
	/d/	07,4	**89**,**2**	00,0	00,0	30,4	08,2	**86**,**4**	00,0	00,0	05,4
	/eh/	25,6	70,6	**00**,**0**	00,0	30,8	03,7	06,5	**77**,**4**	12,4	00,0
	/ih/	13,5	68,6	00,0	**00**,**0**	17,9	09,1	10,3	22,9	**57**,**7**	00,0
	/jh/	00,0	21,2	00,0	00,0	**78**,**8**	00,2	28,3	00,0	00,2	**71**,**3**
		Promedio: 44,68					Promedio: **68**,**32**				

Tabla 3.6. Matrices de confusión: porcentajes de clasificación promedio sobre diez particiones de datos.

banda y el solapamiento de los filtros es mayor para los BFE que para el MFB. Esto significa que los coeficientes de energía resultantes contienen más información redundante, y la CT probablemente no es suficiente para decorrelacionar los CCE. De hecho, se ha realizado un estudio comparando la dependencia estadística de los MFCC y los CCE, obteniendo que los coeficientes optimizados muestran, en general, mayor correlación.

La Figura 3.12 muestra las matrices de correlación de 10 coeficientes cepstrales calculados sobre 1500 tramos de voz. Para realizar esta comparación, se consideró un MFB de 18 filtros, la misma cantidad que en el BFE C4. Los coeficientes de correlación correspondientes al MFB se muestran arriba, y abajo los correspondientes al banco de filtros optimizado C4. Como se puede observar, las matrices de correlación muestran mayor dependencia estadística entre los coeficientes cepstrales correspondientes a los fonemas /eh/ e /ih/, y ésto es mucho más notable para el caso del BFE. Para obtener una medida de la dependencia estadística, se calculó la suma de los coeficientes de correlación para cada fonema. Dichas medidas se pueden observar en la Tabla 3.7, y fueron calculados como $\sum_i \sum_j |m_{i,j}| - \text{trace}(|M|)$, donde los $m_{i,j}$ son los elementos de la matriz de coeficientes de correlación, M. A partir de estos valores también podemos observar que los CCE son más correlacionados que los MFCC para el conjunto de fonemas considerado.

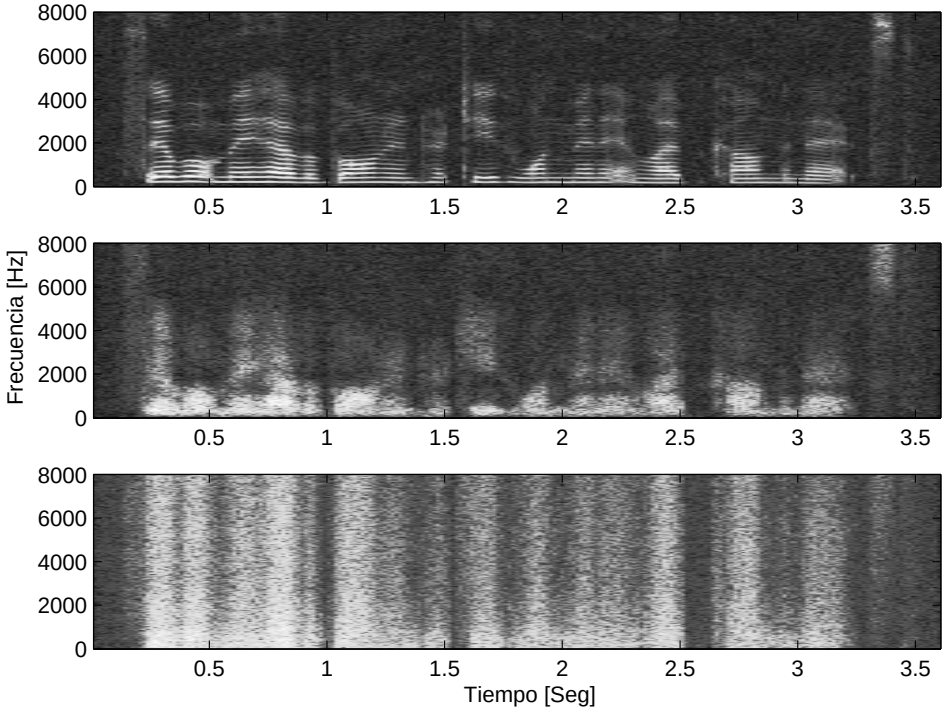


Figura 3.10. Espectrogramas para la frase `SI648` del corpus TIMIT con ruido blanco a 50 dB SNR. Calculado a partir de la señal original (arriba), reconstruido a partir de los MFCC (medio), y reconstruidos a partir de los CCE (abajo).

La dependencia estadística presente en los CCE significa que las mezclas de Gaussianas con MCD no son la mejor opción para modelar las densidades de observación. Esto explica las mejoras obtenidas en los resultados al utilizar mezclas de Gaussianas con MCC en los HMM para modelar las funciones de densidad de observación, en las evaluaciones de aptitud, durante la optimización. Más aún, como los MFCC tampoco están completamente decorrelacionados, la modelización con mezclas de Gaussianas de MCC también permitió mejorar los resultados de ésta representación (ver Tabla 3.1).

3.6.2. Codificación mediante splines

En las distintas corridas del EA se consideraron distintas combinaciones para los valores de los parámetros. Sin embargo, la configuración que permitió ob-

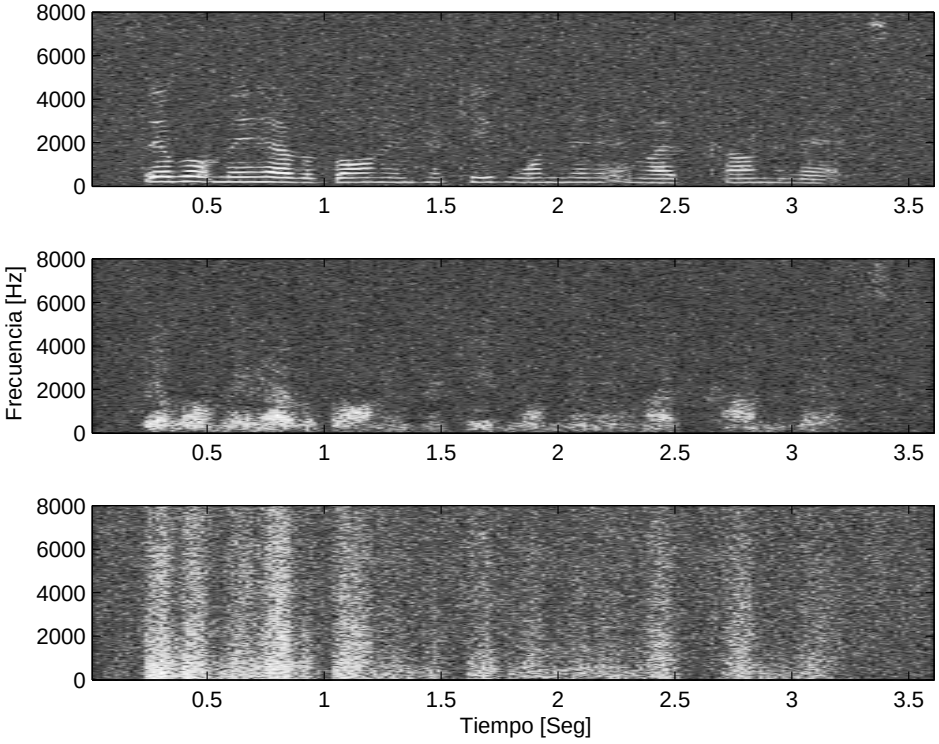


Figura 3.11. Espectrogramas para la frase `SI648` del corpus TIMIT con ruido blanco a 10 dB SNR. Calculado a partir de la señal original (arriba), reconstruido a partir de los MFCC (medio), y reconstruidos a partir de los CCE (abajo).

tener mejores resultados consistió en una población de sólo 30 individuos, una tasa de cruza de 0,9 y una tasa de mutación de 0,07. El parámetro a, discutido anteriormente, fue fijado experimentalmente en 0,1. Este valor ofrece un buen compromiso entre la cantidad de splines válidos (inferior al 15 %) y la variabilidad de los mismos. Durante la optimización, para las evaluaciones de aptitud, se utilizó un conjunto variable de 1000 señales (ocurrencias de los fonemas elegidos) para el entrenamiento y un conjunto variable de 400 señales para prueba. Ambos conjuntos estaban balanceados con respecto a los fonemas y fueron cambiados en cada generación. El re-muestreo del conjunto de entrenamiento fue realizado aleatoriamente a partir de un conjunto de 5000 señales, y en el caso del conjunto de prueba se realizó teniendo en cuenta los errores de clasificación, como se explicó en la Sección 3.2, a partir de un conjunto de 1500 señales. Al finalizar una

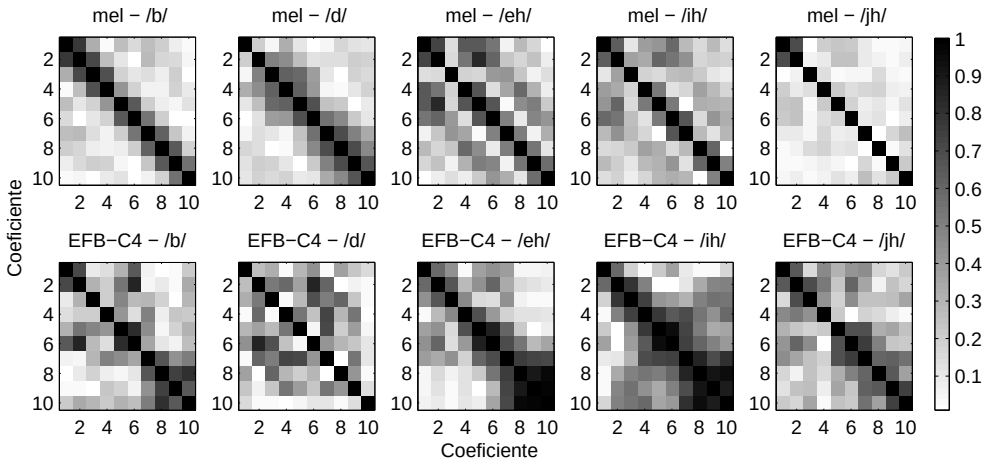


Figura 3.12. Matrices de correlación para los MFCC (arriba) y los CCE (abajo).

	/b/	/d/	/eh/	/ih/	/jh/
MFB	20,9	24,9	30,4	27,2	11,2
BFE-C4	28,8	27,5	33,1	45,5	32,2

Tabla 3.7. Sumas de los coeficientes de correlación.

optimización, se eligieron los bancos de filtros con mejor aptitud.

Con los bancos de filtros seleccionados se realizaron pruebas de validación con diez particiones de datos diferentes, cada una de las cuales estaba constituida por 2500 señales de entrenamiento y 500 señales de prueba. En este caso se realizaron dos tipos de validaciones: en la primera, como en el caso de la sección anterior, se utilizaron señales de entrenamiento limpias y señales de prueba con ruido en distintas cantidades (MMTT); y, en la segunda, las condiciones de ruido de las señales de entrenamiento y de prueba eran las mismas (MTT, del inglés *Match Training and Test*). A partir de los resultados de estas validaciones se seleccionaron los mejores bancos de filtros, descartando aquellos que fueron sobre-optimizados (aquellos que dieron mayor aptitud pero menor resultado de validación). Los resultados promedio de validación de los mejores BFE fueron comparados con los del MFB clásico, bajo las mismas condiciones de entrenamiento y prueba. En los distintos experimentos, para la evaluación de los indi-

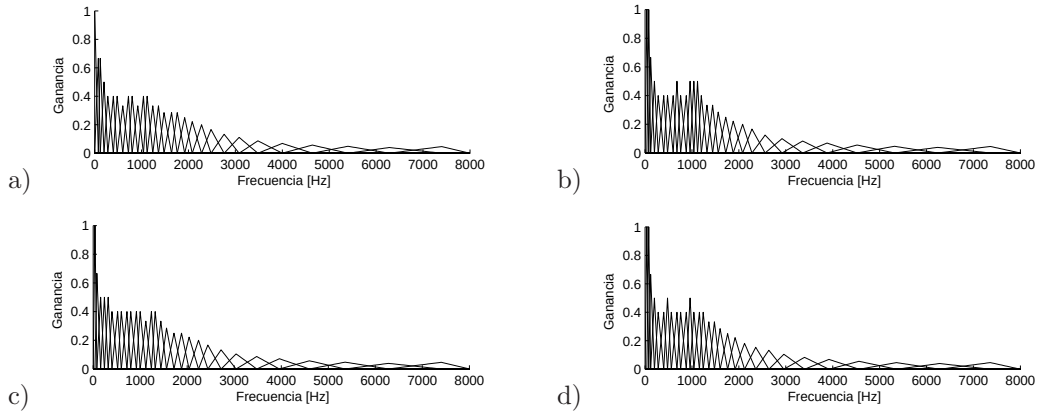


Figura 3.13. Bancos de filtros obtenidos en la optimización de la posición de los filtros (ganancias normalizadas según los anchos de banda) usando señales limpias. a) BFE D1, b) BFE D2, c) BFE D3 y d) BFE D4.

BF	n_f	n_c	MTT					MMTT			
			0 dB	10 dB	20 dB	30 dB	Limpio	0 dB	10 dB	20 dB	30 dB
BFE-D1	30	16	73,14	78,06	73,54	70,74	70,94	23,86	44,06	69,66	70,54
BFE-D2	30	16	73,36	77,94	73,52	71,60	71,16	22,98	43,14	70,52	71,40
BFE-D3	30	16	**73,60**	**78,08**	**73,36**	**71,14**	**71,00**	23,62	44,14	**69,94**	**71,28**
BFE-D4	30	16	72,88	**78,04**	**73,56**	**71,46**	**71,92**	23,68	43,80	**70,06**	**71,28**
MFB	30	16	73,44	77,88	71,22	70,20	69,94	23,72	**44,74**	66,60	70,38

Tabla 3.8. Promedios de los resultados de validación en reconocimiento de fonemas. Bancos de filtros obtenidos en la optimización de la posición de los filtros, mientras que las ganancias se escalan según los anchos de banda, y usando señales limpias.

viduos durante la evolución, el clasificador siempre fue evaluado en condiciones MTT.

Optimización de las frecuencias centrales

En los primeros experimentos, para simplificar el problema, sólo se optimizaron las frecuencias centrales de los filtros. Como se explicó en la sección anterior, en este caso, los cromosomas consisten en sólo 4 genes. Como la ganancia de cada filtro no se optimiza, al igual que en los MFCC, la amplitud de cada filtro es escalada de acuerdo a su ancho de banda. De ésta manera, el área es constante para todos los filtros. Se consideraron bancos de 30 filtros, mientras que el vector

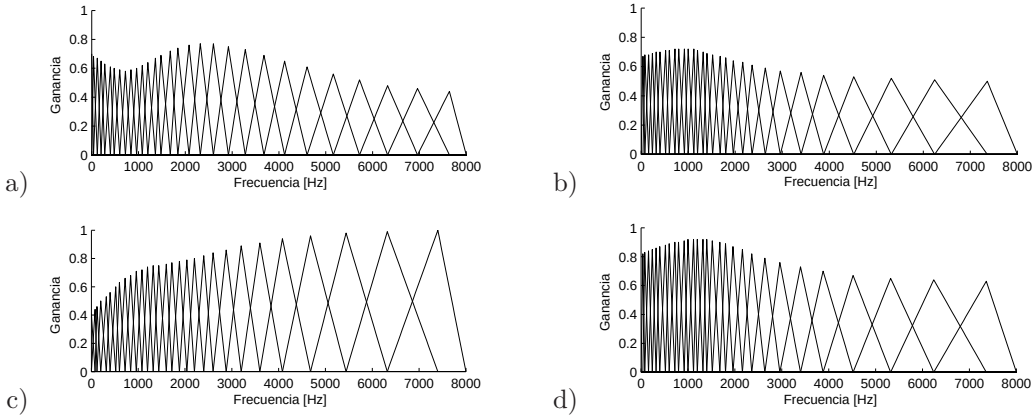


Figura 3.14. Bancos de filtros obtenidos en la optimización simultánea de la posición y la amplitud de los filtros, utilizando señales limpias. a) BFE-E1, b) BFE-E2, c) BFE-E3 y d) BFE-E4.

de características resultante de aplicar la DCT a la secuencia de energías consistía en 16 coeficientes. Debe tenerse en cuenta que el tamaño de los bancos de filtros no está relacionado al tamaño de los cromosomas. La Tabla 3.8 muestra los resultados de validación correspondiente a los BFE D1, D2, D3 y D4, obtenidos de la optimización en la cual se emplearon señales limpias para el entrenamiento y la prueba del clasificador. En las diferentes condiciones de ruido, el desempeño de los BFE se compara con el del banco de filtros clásico. Como se puede observar, en la mayoría de los casos el desempeño de los BFE es mejor que el del MFB, especialmente en condiciones de MTT. La Figura 3.13 muestra estos cuatro BFE, los cuales presentan muy pocas diferencias entre sí. Más aún, la distribución de los filtros es similar a la del clásico MFB. Sin embargo, la resolución que proveen éstos bancos de filtros por debajo de los 2 kHz es mayor, probablemente debido a que ésta es la zona donde se ubican las primeras dos formantes.

En comparación con el caso en que se utilizaron funciones polinomiales para codificar los parámetros [Charbuillet et al., 2007], los bancos obtenidos en dicho caso no eran regulares y no siempre cubrían por completo la banda de frecuencias de interés. Este problema puede atribuirse a la compleja relación que resultaba entre los parámetros de los bancos de filtros y los polinomios optimizados.

BF	n_f	n_c	MTT					MMTT			
			0 dB	10 dB	20 dB	30 dB	limpio	0 dB	10 dB	20 dB	30 dB
BFE-E1	30	16	73,06	78,40	78,56	75,52	74,16	22,94	45,70	55,44	71,80
BFE-E2	30	16	**73,76**	**78,38**	**79,08**	**76,26**	**74,84**	**24,26**	**50,16**	64,84	**73,10**
BFE-E3	30	16	73,54	77,60	78,04	76,02	74,28	22,56	47,32	63,82	70,60
BFE-E4	30	16	**73,74**	**78,74**	**79,18**	**75,66**	**75,40**	23,22	**51,46**	66,58	**72,96**
MFB	30	16	73,44	77,88	71,22	70,20	69,94	23,72	44,74	**66,60**	70,38

Tabla 3.9. Promedios de los resultados de validación en reconocimiento de fonemas. Bancos de filtros obtenidos en la optimización simultánea de la posición y la amplitud de los filtros, utilizando señales limpias.

Optimización de las amplitudes y las frecuencias centrales

El segundo experimento difiere del primero solamente en que la amplitud o ganancia de los filtros también fue optimizada. En este caso, por lo tanto, los parámetros correspondientes a dos splines (uno para controlar la posición y otro para controlar la amplitud de cada filtro) se codificaron en cromosomas de 8 genes en total.

Los resultados de validación para los bancos de filtros BFE-E1, BFE-E2, BFE-E3 y BFE-E4 se muestran en la Tabla 3.9, a partir de los cuales se pueden notar considerables mejoras por sobre el banco de filtros clásico. Con cada uno de los bancos de filtros optimizados se obtuvo mejor desempeño en la mayoría de las condiciones de prueba. Sobre todo, en los casos de MTT en 20 dB, 30 dB y limpio, y para el caso de MMTT de 10 dB, donde las mejoras son más notables. Estos cuatro BFE, que se pueden observar en la Figura 3.14, difieren del MFB (mostrado en la Figura 2.9) en la amplitud de los filtros de altas frecuencias. Es decir, estos bancos de filtros optimizados enfatizan las componentes de altas frecuencias. Al igual que en el caso de los de la Figura 3.13, estos BFE muestran mayor densidad de filtros en el rango de 0 a 2 kHz, comparado con el MFB.

En el tercer experimento, al igual que en el anterior, tanto las posiciones frecuenciales como las amplitudes de los filtros fueron optimizadas. Sin embargo, en este caso se utilizaron señales con ruido a 0 dB SNR para entrenar y probar el clasificador durante la evolución. Los resultados de validación de la Tabla 3.10 revelan que para el caso de 0 dB SNR, en ambas condiciones MTT y MMTT, estos BFE permiten obtener mejores resultados que los de las Tablas 3.8 y 3.9. Sin embargo, los bancos de filtros optimizados con señales limpias obtienen mejor desempeño para la mayoría de las diferentes condiciones de ruido.

Todos los BFE presentados en esta sección son más regulares en comparación

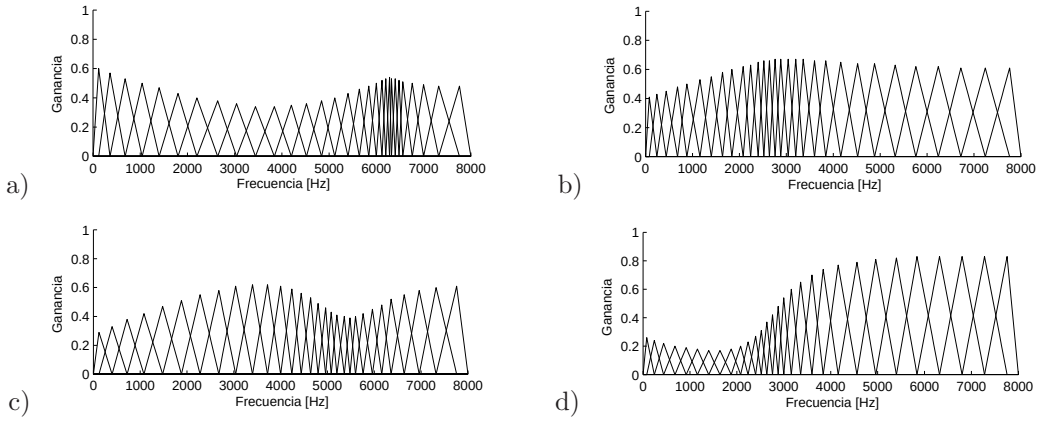


Figura 3.15. Bancos de filtros obtenidos en la optimización simultánea de la posición y la amplitud de los filtros, utilizando señales sucias con SNR de 0 dB. a) BFE-F1, b) BFE-F2, c) BFE-F3 y d) BFE-F4.

BF	n_f	n_c	MTT					MMTT			
			0 dB	10 dB	20 dB	30 dB	limpio	0 dB	10 dB	20 dB	30 dB
BFE-F1	30	16	**73,88**	76,50	**76,24**	**70,78**	69,14	**31,76**	44,46	49,16	67,20
BFE-F2	30	16	**74,66**	**78,60**	**78,96**	**73,78**	**70,76**	**25,74**	**46,68**	49,76	66,88
BFE-F3	30	16	**74,90**	77,18	**76,10**	**70,56**	69,48	**29,70**	44,50	49,40	68,06
BFE-F4	30	16	**74,76**	**78,16**	**78,54**	**75,36**	**71,04**	**24,80**	**46,08**	52,12	66,36
MFB	30	16	73,44	77,88	71,22	70,20	69,94	23,72	44,74	**66,60**	**70,38**

Tabla 3.10. Promedios de los resultados de validación en reconocimiento de fonemas. Bancos de filtros obtenidos en la optimización de la posición y las amplitudes de los filtros, utilizando señales sucias.

con los que se presentaron en la sección anterior, lo que sugiere que con esta alternativa se obtiene una convergencia es más estable. Si bien los resultados presentados en la sección anterior muestran algunas mejoras, la forma de los bancos de filtros optimizados no es fácil de explicar. Además, bancos de filtros muy diferentes entre sí permitieron obtener resultados comparables, sugiriendo que se estaba tratando con un problema mal condicionado. En este caso, en el que se realiza la codificación mediante splines, sólo se observan disimilaridades entre los bancos de filtros optimizados con señales sucias.

En la Figura 3.15 se puede observar que los bancos de filtros evolucionados con señales ruidosas difieren ampliamente del MFB y de los que fueron optimiza-

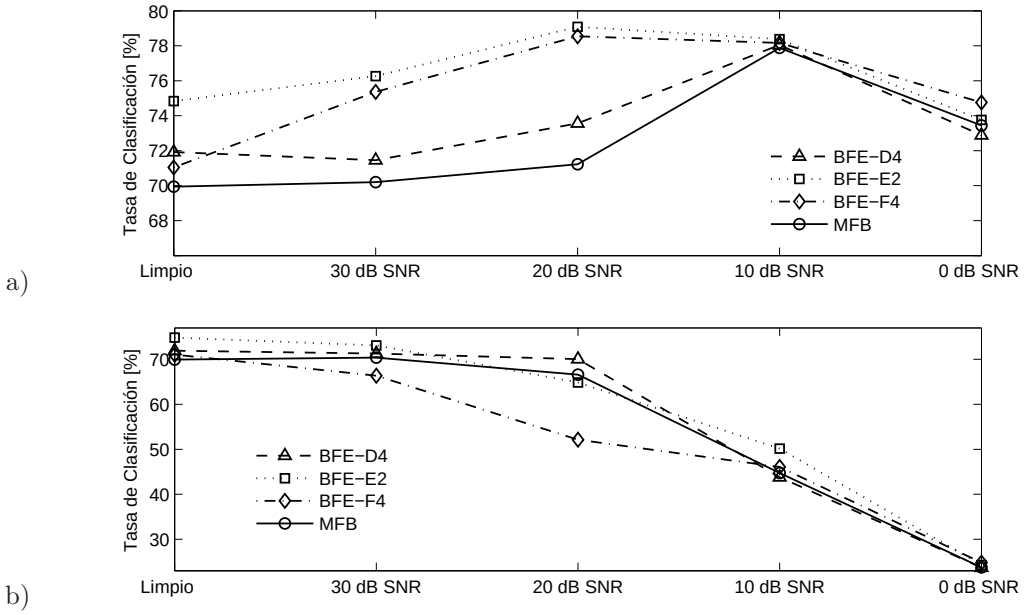


Figura 3.16. Resultados promedio de validación obtenidos en la clasificación de fonemas comparando MFB con BFE-D4, BFE-E2 y BFE-F4 en distintas condiciones de ruido. a) validación en condiciones de MTT, y b) validación en condiciones de MMTT.

dos con señales limpias. Por ejemplo, en los BFE de la Figura 3.15 la densidad de filtros es mayor frecuencias altas. Además, en contraste con los BFE anteriores, en este caso el escalado de las amplitudes de los filtros da menos importancia a las bandas de bajas frecuencias. Esta característica, que está presente en todos estos BFE y los diferencia del MFB, les permite dar mayor importancia a frecuencias formantes más altas. Sin embargo, las notables disimilitudes entre estos cuatro BFE sugieren que la optimización resulta mucho más compleja cuando se utilizan señales ruidosas [Vignolo et al., 2011b].

La Figura 3.16 resume los resultados de las Tablas 3.8, 3.9 y 3.10 comparando los BFE -D4, -E2 y -F4 con el MFB en diferentes condiciones de ruido. En la Figura 3.16(a) se puede observar que, en condiciones de MTT, los BFE permiten obtener mejores resultados que el MFB para la mayoría de los casos. Además, la Figura 3.16(b) muestra ciertas mejoras de los BFEs -D4 y -E2 sobre el MFB en el caso de MMTT.

La Tabla 3.11 muestra algunas matrices de confusión para la clasificación de fonemas a partir del MFB y el BFE-E2, obtenidas en la validación en distintas

		MFB (30/16)					BFE-E2				
		/b/	/d/	/eh/	/ih/	/jh/	/b/	/d/	/eh/	/ih/	/jh/
10 dB	/b/	**80,0**	15,1	01,1	02,9	00,9	**81,3**	15,2	00,5	02,2	00,8
	/d/	20,1	**72,2**	00,2	02,0	05,5	20,4	**71,0**	00,6	01,9	06,1
	/eh/	03,0	01,0	**78,4**	17,6	00,0	02,2	01,2	**81,6**	15,0	00,0
	/ih/	02,0	03,2	21,3	**73,2**	00,3	01,5	01,1	23,9	**73,1**	00,4
	/jh/	00,0	14,3	00,0	00,1	**85,6**	00,5	14,5	00,0	00,1	**84,9**
		Promedio: 77,88					Promedio: **78,38**				
20 dB	/b/	**74,1**	21,5	02,2	10,7	00,5	**79,8**	16,7	00,7	02,1	00,7
	/d/	15,0	**78,8**	00,9	10,4	03,9	17,9	**74,8**	00,6	02,8	03,9
	/eh/	12,7	04,9	**55,6**	26,5	00,3	00,7	01,0	**76,6**	21,7	00,0
	/ih/	06,3	03,9	27,1	**62,4**	00,3	00,4	00,5	24,0	**75,1**	00,0
	/jh/	00,7	13,6	00,0	00,5	**85,2**	00,5	09,9	00,1	00,4	**89,1**
		Promedio: 71,22					Promedio: **79,08**				
30 dB	/b/	**53,2**	32,2	06,9	07,0	00,7	**78,9**	18,6	01,0	01,0	00,5
	/d/	11,0	**77,0**	02,7	04,4	04,9	17,1	**76,5**	00,8	01,3	04,3
	/eh/	01,3	02,3	**68,9**	27,4	00,1	02,3	01,0	**72,1**	24,6	00,0
	/ih/	00,9	01,9	30,2	**66,9**	00,1	01,8	01,3	26,3	**70,6**	00,0
	/jh/	01,5	12,1	00,5	00,9	**85,0**	00,7	14,8	00,2	01,1	**83,2**
		Promedio: 70,2					Promedio: **76,26**				
limpio	/b/	**54,4**	28,9	07,9	07,8	01,0	**74,9**	18,9	02,4	03,3	00,5
	/d/	12,2	**76,3**	01,9	04,8	04,8	15,5	**78,1**	00,9	01,0	04,5
	/eh/	02,2	02,1	**69,4**	26,0	00,3	01,4	01,3	**67,9**	29,3	00,1
	/ih/	02,4	01,5	31,8	**64,2**	00,1	03,1	01,3	26,7	**68,9**	00,0
	/jh/	02,1	11,7	00,2	00,6	**85,4**	01,1	13,2	00,9	00,4	**84,4**
		Promedio: 69,94					Promedio: **74,84**				

Tabla 3.11. Matrices de confusión obtenidas en promedio sobre diez particiones de datos en condiciones de MTT para MFB y BFE-E2. Porcentajes de acierto

condiciones de ruido y para el caso de MTT. A partir de estas matrices, se puede notar que los fonemas /b/, /eh/ y /ih/ son confundidos frecuentemente con el MFB, pero se obtienen tasas de acierto significativamente mejores cuando se utiliza BFE-E2 en la representación. Además, con BFE-E2 se reduce la varianza entre las tasas de acierto para los distintos fonemas. También se puede notar que el fonema /b/ es confundido mayormente con el fonema /d/ y viceversa, y lo mismo ocurre con las vocales /eh/ y /ih/. Esto sucede con ambos bancos de filtros MFB y BFE-E2, sin embargo, el banco de filtros optimizado permite reducir estas confusiones considerablemente [Vignolo et al., 2011b].

Para analizar qué información es la que éstos bancos de filtros permiten con-

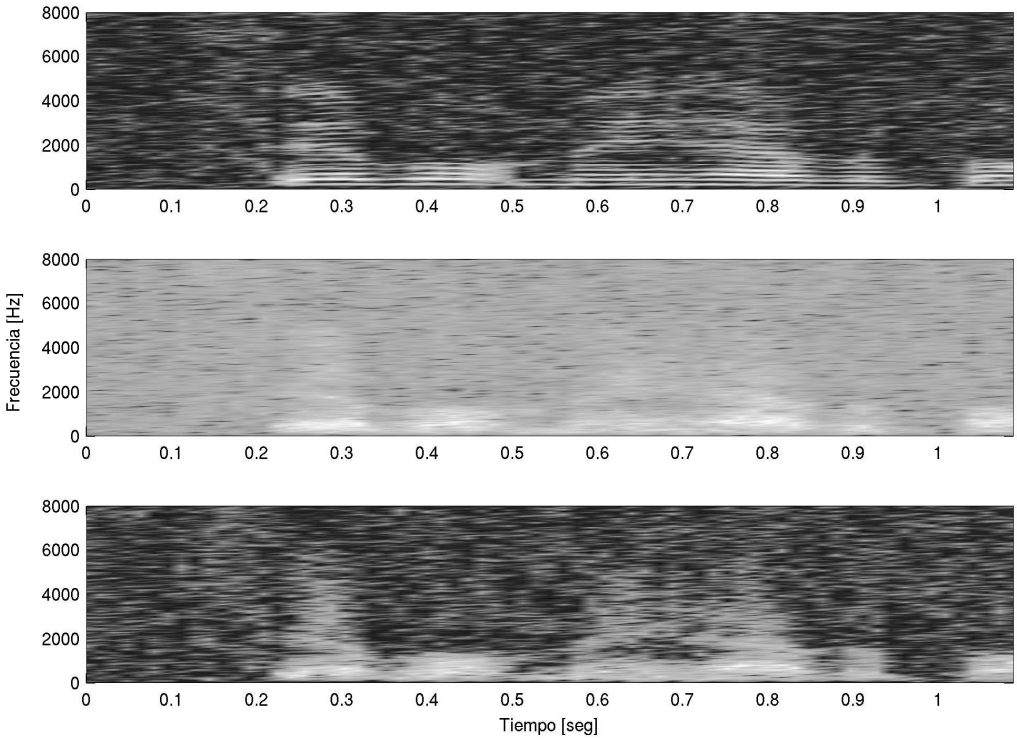


Figura 3.17. Espectrogramas de un fragmento de la frase `SI648` del corpus TIMIT con ruido blanco aditivo en SNR de 20 dB. Calculado a partir de la señal original (arriba), reconstruido a partir de los MFCC (medio) y reconstruido a partir de los coeficientes obtenidos con el BFE-E4.

servar, al igual que en la Sección 3.6.1, se reconstruyó una estimación del espectro de magnitud de tiempo corto de la señal de voz [Ellis, 2005]. Los espectrogramas para un fragmento de la frase `SI648` del corpus TIMIT, con ruido blanco aditivo a 20 dB SNR, se pueden observar en la Figura 3.17. El espectrograma de la parte superior corresponde al de la señal original, en el centro se muestra el espectrograma reconstruido a partir de los MFCC, y debajo de éste se muestra el espectrograma reconstruido a partir de los CCES calculados mediante el BFE-E4. Se puede notar que el espectrograma reconstruido a partir de los CCES está menos afectado por el ruido que los otros dos. Además, la información de las frecuencias formantes es realzada por el BFE-E4, facilitando la clasificación de los fonemas. Esto significa que la distribución de los filtros y los anchos de bandas del BFE-E4 permiten conservar más información relevante en comparación con

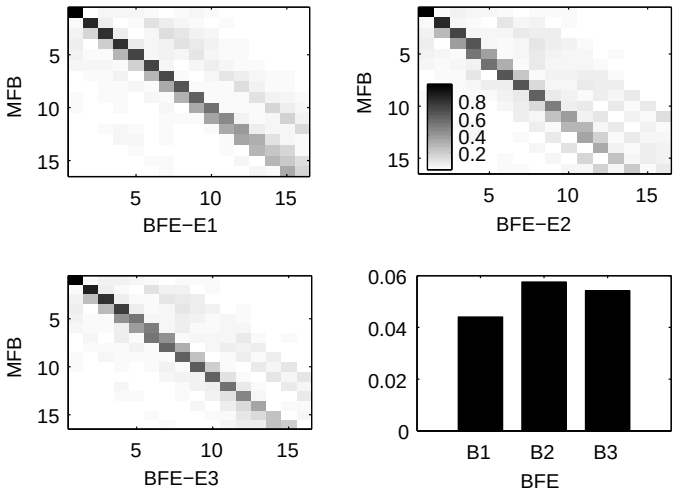


Figura 3.18. Coeficientes de correlación de Pearson al cuadrado, entre los MFCC y los CCES obtenidos con BFE-E1, BFE-E2 y BFE-E3 (arriba a la izquierda, arriba a la derecha y abajo a la izquierda, respectivamente). Suma normalizada de los coeficientes de correlación fuera de la diagonal (abajo a la derecha).

el MFB.

Para evaluar la relación entre los MFCC y los CCES, se realizó la comparación mediante las coeficientes de correlación de Pearson r. La Figura 3.18 muestra las matrices de correlación comparando los MFCC con los CCES (obtenidos mediante los BFE -E1, -E2 y -E3) sobre 17846 tramos de fonemas con ruido blanco aditivo a 0 dB SNR. Se observa que aproximadamente la primer mitad de los coeficientes están altamente correlacionados entre los bancos de filtros bajo comparación. Además, en el caso del BFE-E2, hay más coeficientes mayores a cero fuera de la diagonal. Esto significa que los CCES obtenidos con el BFE-E2 son los que están menos correlacionados con los MFCC, ya que la información es distribuida de manera diferente entre los distintos coeficientes. Esto se puede apreciar mejor en el gráfico de barras, en el cual la altura de las barras está dada por la suma normalizada de los coeficientes de correlación fuera de la diagonal. Se debe tener en cuenta que el BFE-E2 es el que permitió obtener los mejores resultados de validación [Vignolo et al., 2011b].

Una análisis similar se realizó entre los coeficientes de un mismo banco de filtros, para evaluar qué tan correlacionados están entre sí. En la Figura 3.19 se muestran las matrices de coeficientes de correlación al cuadrado para los MFCC

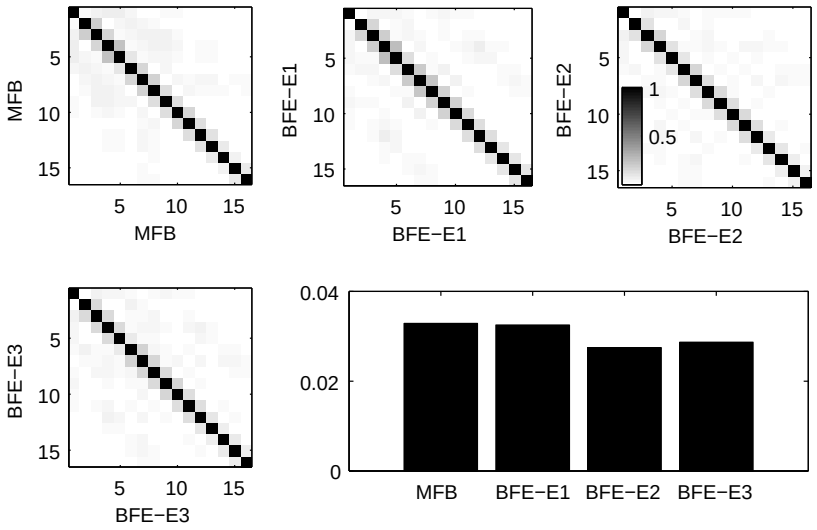


Figura 3.19. Coeficientes de correlación de Pearson al cuadrado de los MFCC, y los CCES obtenidos con BFE-E1, BFE-E2 y BFE-E3 (arriba a la izquierda, arriba a la derecha y abajo a la izquierda, respectivamente). Suma normalizada de los coeficientes de correlación fuera de la diagonal (abajo a la derecha).

y los CCES (BFE-E1, BFE-E2 y BFE-E3). Se puede notar que la matriz para el BFE-E2 es la que tiene menos cantidad de coeficientes distintos de cero fuera de la diagonal. Además, la suma normalizada de los coeficientes fuera de la diagonal es menor para este BFE, lo que significa que estos CCES son menos correlacionados que los MFCC. Por esta razón los CCES obtenidos con el BFE-E2 satisfacen mejor los supuestos de independencia para el uso de los HMM, ya que es práctica común emplear mezclas de Gaussianas con matrices de covarianza diagonales en la modelización de las densidades de observación [Demuynck et al., 1998].

3.6.3. Resumen comparativo

La Tabla 3.12 resume los resultados de validación para los mejores bancos de filtros obtenidos mediante codificación directa y codificación mediante splines, y los compara con los bancos de filtros de referencia. Como se puede ver, para los distintos niveles de ruido considerados, los bancos de filtros evolucionados permiten obtener mejores resultados de clasificación. En el caso de los bancos de filtros optimizados con la estrategia de codificación directa, éstos obtienen

BF	n_f	n_c	0 dB	10 dB	20dB	30 dB	Limpio
BFE-A3	29	15	**32,92**	46,36	66,08	65,04	64,32
BFE-A6	21	11	**35,78**	46,88	61,52	60,36	60,80
BFE-B6	18	10	**31,92**	**67,20**	68,44	67,20	66,64
BFE-B7	19	10	**31,98**	44,42	64,44	67,58	67,26
BFE-C4	18	10	**36,88**	**68,32**	69,82	67,42	68,08
BFE-C5	18	10	**30,42**	44,32	65,14	63,52	63,40
BFE-D3	30	16	23,62	44,14	69,94	**71,28**	**71,00**
BFE-D4	30	16	23,68	43,80	**70,06**	**71,28**	**71,92**
BFE-E2	30	16	24,26	50,16	64,84	**73,10**	**74,84**
BFE-E4	30	16	23,22	51,46	66,58	**72,96**	**75,40**
BFE-F2	30	16	25,74	46,68	49,76	66,88	**70,76**
BFE-F4	30	16	24,80	46,08	52,12	66,36	**71,04**
MFB	23	13	23,18	44,68	68,40	69,80	69,16
MFB	30	16	23,72	44,74	66,60	70,38	69,94
HFCC	30	13	25,98	62,78	**70,54**	69,42	70,36
Slaney	40	20	30,28	54,76	62,02	61,52	62,78

Tabla 3.12. Porcentajes de clasificación obtenidos con fonemas del inglés. Promedio sobre diez particiones de datos y en condiciones MMTT

mejores resultados que el MFB, HFCC y Slaney para el caso de señales sucias. En cambio, para señales con menor cantidad de ruido (de 20 dB SNR a limpio) los bancos de filtros que obtienen mejores resultados son los que se obtuvieron mediante la estrategia de codificación con splines. Por otro lado, si tenemos en cuenta como referencia solamente el banco de filtros más comúnmente utilizado (MFB), podemos ver que la comparación favorece mucho más a varios de los BFE. Para facilitar la comparación, los resultados más destacados de la Tabla 3.12 se pueden observar también en la Figura 3.20. Se puede ver, con claridad, que distintos BFE logran mejorar los resultados de los métodos de referencia.

Como estos bancos de filtros se optimizaron para un conjunto reducido de fonemas, no se puede esperar a priori una mejora en los resultados de reconocimiento continuo. Sin embargo, se realizaron pruebas preliminares utilizando un BFE en un sistema de reconocimiento completo y se obtuvieron resultados alentadores. Para ello, se construyó un sistema de reconocimiento utilizando herramientas de HTK, y el desempeño obtenido con los CCES fue comparado con el desempeño de la representación clásica de MFCC, utilizando frases del corpus TIMIT (región 1)

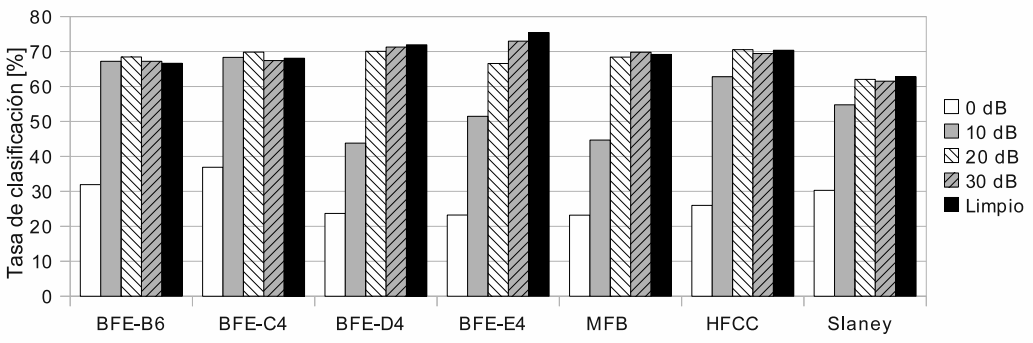


Figura 3.20. Porcentajes de clasificación obtenidos con fonemas del inglés. Promedio sobre diez particiones de datos y en condiciones MMTT

con diferentes cantidades de ruido blanco aditivo (en condiciones MMTT). Como pre-procesamiento de las señales, antes de calcular los MFCC/CCES, se realizó el pre-énfasis en cada tramo de voz. Luego, a los MFCC y CCES obtenidos se les añadieron los coeficientes delta y aceleración, para incluir información temporal en la representación. Las tasas de reconocimiento de frases y palabras resultaron cercanas para los MFCC y los CCES en casi todos los casos. En el caso de SNR de 15 dB, las tasas de reconocimiento obtenidas fueron de 15,83 % y 31,98 % para los bancos de filtros MFB y BFE-E4, respectivamente. Esto sugiere que, aún cuando la optimización se realiza sobre un conjunto reducido se fonemas, los CCES obtenidos de esta manera siguen siendo una buena representación para los fonemas que no son considerados. Además, debe tenerse en cuenta que los cinco fonemas considerados representan solamente el 9,38 % (`/b/`: 1,49 % , `/d/`: 2,28 %, `/eh/`: 2,35 %, `/ih/`: 2,76 %, `/jh/`: 0,51 %) de la cantidad total de ocurrencias de fonemas en las frases de prueba. Es decir, de un total de 3956 patrones, sólo 371 corresponden a los cinco fonemas considerados para la optimización.

Otro aspecto a tener en cuenta es la carga computacional de las optimizaciones descriptas. Una corrida de 2500 generaciones (que es la cantidad de generaciones en los últimos experimentos descriptos) toma aproximadamente 84 horas (cerca de 2 minutos por cada generación) en un cluster de computadoras constituido por once procesadores de 3 GHz de velocidad de reloj. La mayor parte de la carga computacional de la optimización se debe a la evaluación de la aptitud de los individuos, es decir, el entrenamiento y la prueba del clasificador basado en HMM. Comparando las dos alternativas de codificación propuestas, debe tenerse en cuenta que la reducción en el tamaño del cromosoma, lograda con la estrategia

de codificación mediante splines, permitió al EA converger a mejores soluciones en tiempos de procesamiento similares. Es importante notar que la propuesta no implica ninguna carga adicional al procedimiento estándar de reconocimiento de voz. El proceso de optimización es una etapa previa al reconocimiento, y puede ser considerado como parte del entrenamiento del reconocedor. Es decir, una vez obtenido un banco de filtros que permita mejorar los resultados, éste se incorpora y queda fijo en el sistema de RAH. Además, las técnicas de extracción de características para obtener los MFCC, los ECC y los CCES son similares, por lo cual no es necesario realizar un cambio importante en la etapa de procesamiento.

Capítulo 4

Paquetes de onditas evolutivos

4.1. Introducción

En el Capítulo 3 se presentó una estrategia basada en algoritmos evolutivos para encontrar una representación más apropiada para la clasificación de fonemas. Esta búsqueda se realizó a partir de la representación clásica de las señales de habla. En este capítulo también se propone una estrategia evolutiva, pero en este caso se optimiza una representación no convencional basada en diccionarios de onditas.

El procesamiento basado en onditas se ha encontrado útil para el análisis de señales no estacionarias, y en los últimos años se ha estudiado su aplicación para la extracción de características de las señales de voz [Wu y Lin, 2009]. La transformada paquete de onditas es una herramienta con características especialmente útiles para el análisis de señales de habla, las cuales tienen ciertas propiedades estacionarias y otras características transitorias [Rufiner, 2009]. Para el caso discreto, el análisis multi-resolución realizado por esta transformada puede ser implementado mediante una estructura de bancos de filtros [Mallat, 1989; Vetterli y Herley, 1992]. Una familia de paquetes de onditas en general no constituye una base ortogonal, y por lo tanto, estos diccionarios sobre-completos proporcionan un conjunto altamente redundante de coeficientes para representar una señal. Debido a la dimensionalidad de la representación obtenida, en general se requiere la selección de una base particular dentro de la familia de bases disponibles. Existe un algoritmo que minimiza una función de entropía de la señal analizada para encontrar la mejor base ortogonal [Wickerhauser, 1991], que resulta útil para aplicaciones de compresión. Por otro lado, para encontrar una base ortogonal adecuada para clasificación de señales, se desarrolló una técnica denominada base

discriminante local [Saito, 1994]. Sin embargo, en problemas de clasificación, no resulta imprescindible que la base utilizada para representar las señales sea ortogonal. Si se quita entonces esta restricción se obtiene en un conjunto redundante de coeficientes, el cual es conveniente optimizar antes de utilizarlo para alimentar un clasificador. Entonces, el problema para diseñar una representación de señales basada en onditas consiste en cómo elegir una base adecuada para cada aplicación particular.

Por ejemplo, un método para elegir una familia de onditas y sus parámetros de manera de facilitar la tarea de reconocimiento de fonemas se propuso en [Rufiner y Goddard, 1997]. En [Farooq y Datta, 2004] se ha propuesto una representación basada en la transformada ondita discreta con ciertas modificaciones para obtener un conjunto de coeficientes robustos al ruido. De manera similar, para la tarea de identificación de hablantes también se han propuesto representaciones robustas basadas en la transformada paquete de onditas [Hsieh et al., 2002]. Además, se han propuesto distintas alternativas para la optimización de descomposiciones basadas en onditas. En [Ray y Chan, 2001] se propuso un método de extracción de características a partir de los coeficientes de la transformada ondita de acuerdo a un criterio de clasificación. Otra alternativa interesante fue propuesta en [Jones et al., 2001], en la cual se emplea un algoritmo genético basado en lenguaje para diseñar onditas con el objetivo de maximizar el desempeño de un clasificador. Por otro lado, también se ha propuesto un método para optimizar descomposiciones sobre-completas a partir de varios diccionarios, empleando técnicas de computación evolutiva y para su aplicación a la aproximación de señales [Ferreira da Silva, 2003]. También se ha propuesto el uso de un GA en la optimización de características basadas en paquete de onditas para detección de patologías en señales de voz, seleccionando los nodos del árbol de descomposición en base a un criterio de entropía [Behroozmand y Almasganj, 2007]. Otros trabajos que proponen métodos evolutivos para optimizar descomposiciones basadas en onditas (aunque no necesariamente en paquetes de onditas) son [Ferreira da Silva, 2001; Lankhorst et al., 2003; Schell y Uhl, 2003]. Sin embargo, la flexibilidad provista por la descomposición completa de la transformada paquete de onditas aún no se ha explotado completamente en la búsqueda de un conjunto de características que permita mejorar los resultados de clasificación. Además, cuando esta búsqueda no está restringida a representaciones no redundantes, existe un gran número de bases no ortogonales posibles, por lo que resulta en un problema combinatorio complejo.

En este capítulo se presenta un método para optimizar una descomposición

basada en onditas a partir de diccionarios sobre-completos. A diferencia de los trabajos citados en el párrafo anterior, en este caso no se impone ninguna restricción sobre los posibles esquemas de descomposición. Esta estrategia consiste en la utilización de un GA para la selección de las componentes de la transformada paquete de onditas que resulten más relevantes para la clasificación [Vignolo y Milone, 2005a]. La propuesta se denomina *paquetes de onditas evolutivos* (POE). Este esquema de selección de características, que se conoce como *wrapper* (o envoltorio) [Hofmann et al., 2004; Yu y Cho, 2006], es muy utilizado actualmente porque permite obtener mejores resultados en comparación con otras técnicas [Kohavi y John, 1997].

Este capítulo se organiza de la siguiente manera. En primer lugar se introduce la transformada ondita discreta y seguidamente se analiza el problema de la resolución temporal-frecuencial que ésta posee, explicando la solución que proveen los paquetes de onditas. Luego se introduce el problema de optimización abordado en esta parte de la tesis y se presenta la estrategia *wrapper* propuesta. Seguidamente se describe el corpus fonético utilizado y se detalla la metodología seguida en la experimentación. Finalmente se presentan y discuten los resultados obtenidos.

4.2. Análisis basado en onditas

En el Capítulo 2 se describió el análisis de la transformada onditas para señales de tiempo continuo. En esta sección se completa dicha revisión para abarcar el caso de las señales de tiempo discreto, introduciendo la transformada ondita discreta y la transformada paquete de onditas, sobre la cual se realiza la optimización propuesta.

4.2.1. Transformada ondita discreta

El diseño de una versión discreta de la transformada ondita consiste esencialmente en definir una apropiada red discreta de parámetros (escalas y traslaciones) de modo que la familia de onditas obtenida sea admisible, es decir, que cumpla con las condiciones mencionadas en la Sección 2.3.3. Existen varias clases de onditas admisibles, como las *Symmlets*, *Coiflets*, *Spline* y *Daubechies*, entre otras ampliamente difundidas en la literatura y en el software disponible. Entre éstas, encontramos diversas variantes, y particularmente las que generan bases ortonormales de onditas.

Sea $\dot{x}(t)$ una señal de tiempo continuo que es muestreada uniformemente a intervalos $T_m = N^{-1}$ en $[0, 1]$. Su transformada ondita se puede calcular únicamente a escalas $N^{-1} < s < 1$. En el caso discreto, es más sencillo normalizar la distancia de muestreo a 1 y considerar la señal dilatada $x(t) = \dot{x}(N^{-1}t)$. Para simplificar la notación, denotamos $x[n] = x(n)$ la señal discreta de N muestras. Para calcular su transformada discreta, se evalúa en las escalas $s = a^j$ con $a = 2^{\frac{1}{v}}$, lo que hace que en cada intervalo $[2^j, 2^{j+1}]$ haya v valores intermedios. La función ondita resulta:

$$\psi_j[n] = \frac{1}{\sqrt{a^j}}\psi\left(\frac{n}{a^j}\right). \tag{4.1}$$

Si denotamos como $\psi_j^*[n]$ el conjugado complejo de $\psi_j[n]$, la *transformada ondita discreta* (DWT, del inglés *discrete wavelet transform*) de $x[n]$ puede escribirse entonces como la convolución circular

$$Wx\left[n, a^j\right] = \sum_{m=0}^{N-1} x[m]\psi_j^*[m-n], \tag{4.2}$$

donde $a^j \in [2N^{-1}, K^{-1}]$ y K es el soporte de ψ (que es distinta de 0 en el intervalo $[-K/2, K/2]$). Es importante notar que en la transformada (4.2) la función base no se especifica explícitamente.

Filtro de escala discreto: La transformada ondita calculada hasta una escala a^j no es una representación completa de la señal. Es necesario agregar las frecuencias bajas $Lx[n, a^j]$ correspondiente a las escalas mayores que a^j. Un filtro de escala periódico se calcula muestreando la función de escala $\phi(t)$ definida en (2.8)

$$\phi_j[n] = \frac{1}{\sqrt{a^j}}\phi\left(\frac{n}{a^j}\right) \text{ para } n \in [-\frac{N}{2}, \frac{N}{2}]. \tag{4.3}$$

Entonces, la aproximación a bajas frecuencias es calculada mediante

$$Lx[n, a^j] = \sum_{m=0}^{N-1} x[m]\phi_j^*[m-n], \tag{4.4}$$

y la fórmula de reconstrucción es obtenida discretizando la integral (2.10). Suponiendo que $a^I = 2$ es la mínima escala, como la transformada discreta se calcula sobre una secuencia exponencial de escalas $\{a^j\}_j$, la reconstrucción aproxima mediante [Mallat, 1999]

$$x[n] \approx \frac{\log_e a}{C_\psi} \sum_{j=1}^{J} \frac{1}{a^j} Wx[n, a^j] \circledast \psi_j[n] + \frac{1}{C_\psi a^j} Lx[n, a^j] \circledast \phi_j[n], \tag{4.5}$$

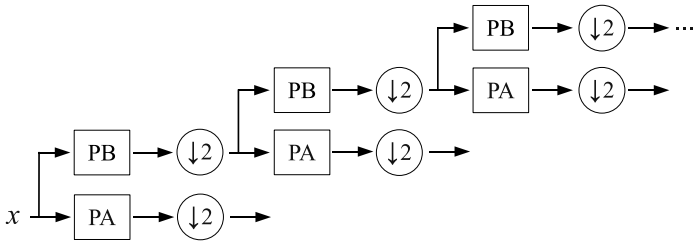


Figura 4.1. Algoritmo para obtener la DWT de una señal x. PA y PB representan un filtro pasa-altos y un filtro pasa-bajos respectivamente.

donde $\circledast$ representa convolución circular.

A diferencia de las series de Fourier, el análisis se realiza por octavas o rangos de frecuencia que duplican su dimensión hacia las altas frecuencias, a la vez que se reduce el rango temporal de localización. Es posible entonces localizar por medio del espectro, tanto fenómenos locales como patrones de auto-similaridad, a distintas escalas. Por supuesto, existen también desventajas. Por un lado, la discretización no conserva ciertas importantes propiedades de la transformada ondita continua (particularmente, la de invarianza respecto de las traslaciones).

Algoritmo de la DWT

Los operadores de traslación y dilatación aplicados a la ondita madre son realizados para calcular los coeficientes ondita, que representan la correlación entre la ondita y una sección localizada de la señal. Los coeficientes son calculados para cada escala y traslación de una ondita, dando una función tiempo-escala relativa a la correlación de las onditas con la señal. Como se puede observar en la Figura 4.1, en el proceso de la DWT la señal se divide en bloques diádicos (tanto la traslación como el escalamiento están basados en potencias de 2).

En el análisis multi-resolución basado en la DWT, una señal se descompone en sucesivos niveles de aproximación (frecuencias bajas) y detalle (frecuencias altas). En cada nivel se obtienen los coeficientes de la transformada aplicando el mismo par de filtros (PA y PB en la Figura 4.1) a la aproximación obtenida en el nivel anterior. El efecto de escalado de la ondita se obtiene en el algoritmo sub-muestreando la señal en cada nivel.

Los filtros empleados generalmente son los de cuadratura espejo (QMF, del

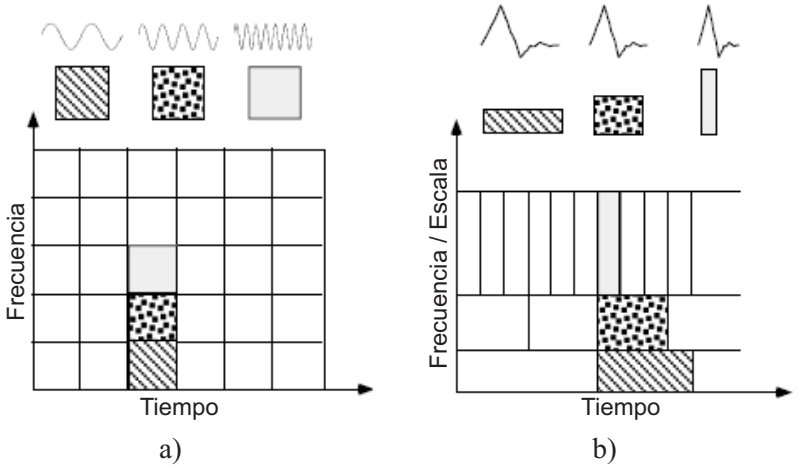


Figura 4.2. Funciones base y resolución tiempo-frecuencia: a) STFT, b) DWT.

inglés *quadrature mirror filter*), que están relacionados por la ecuación

$$g[L-1-n] = (-1)^n h[n], \tag{4.6}$$

donde $g[n]$ es el filtro pasa-altos, $h[n]$ es el filtro pasa-bajos y L es la longitud en muestras del filtro que se utilizará en la transformación. La salida de los filtros se puede expresar mediante las convoluciones discretas

$$y_g[m] = \sum_n x[n]g[-n+2m], \tag{4.7}$$

$$y_h[m] = \sum_n x[n]h[-n+2m], \tag{4.8}$$

donde y_g y y_h son las salidas de los filtros pasa-altos y pasa-bajos respectivamente. De este modo se puede analizar una señal continua que previamente fue muestreada para hacerla discreta. El proceso se puede revertir mediante la fórmula

$$x[n] = \sum_{m=-\infty}^{\infty} \left(y_g[m]g[-n+2m] + y_h[m]h[-n+2m] \right), \tag{4.9}$$

en la cual se obtiene la reconstrucción por medio de las salidas de los filtros y sus respectivas respuestas al impulso, considerando el proceso de sub-muestreo y supra-muestreo.

La información obtenida mediante esta transformada se representa en un escalograma, que es el equivalente al espectrograma en el análisis mediante onditas. En la Figura 4.3 se pueden apreciar los escalogramas obtenidos mediante la CWT y la DWT de una señal de habla en español.

4.2.2. Transformada paquete de onditas

La transformada paquete de onditas (WPT, del inglés *wavelet packet transform*) es una generalización de la descomposición onditas que ofrece un rango más amplio de posibilidades para el análisis de señales.

En el análisis de la transformada ondita una señal se descompone en aproximación y detalle. Luego la aproximación se vuelve a descomponer en un segundo nivel de aproximación y detalle, y el proceso se repite (Figura 4.1). Utilizando el razonamiento seguido por Wickerhauser [Hess-Nielsen y Wickerhouser, 1996] se pueden descomponer también los componentes de alta frecuencia (detalles) así como los componentes de baja frecuencia (aproximaciones). Esto produce lo que se llama árbol de descomposición de la WPT (Figura 4.5).

La raíz dicho árbol es la señal original, el siguiente nivel es el resultado de un paso de la transformada. Los niveles subsiguientes en el árbol se construyen aplicando la transformada recursivamente al resultado obtenido mediante los filtrados pasa-altos y pasa-bajos del paso anterior.

Para una función ondita dada, se genera una colección de bases llamada paquete de onditas. Cada una de estas bases ofrece una manera particular de codificar señales, preservando la energía global y con posibilidad de reconstrucción exacta. Los átomos de un paquete de onditas son funciones bien localizadas tanto en tiempo como en frecuencia, como por ejemplo, una nota musical. Estos átomos son indexados por tres parámetros: posición, escala (como en la CWT), y frecuencia. El primero y el tercero se pueden tomar de los centros de masa de $|\psi|^2$ y $|\hat{\psi}|^2$, donde $\hat{\psi}$ es la transformada de Fourier de ψ. El segundo parámetro se puede ver como el ancho característico de $|\psi|^2$ o lo que es equivalente, la incerteza en la posición. Por el principio de Heisenberg, es también recíproco de la incerteza en la frecuencia.

Siguiendo el mismo razonamiento introducido para la DWT (Sección 4.2.1), pueden descomponerse también los detalles (componentes de altas frecuencias)

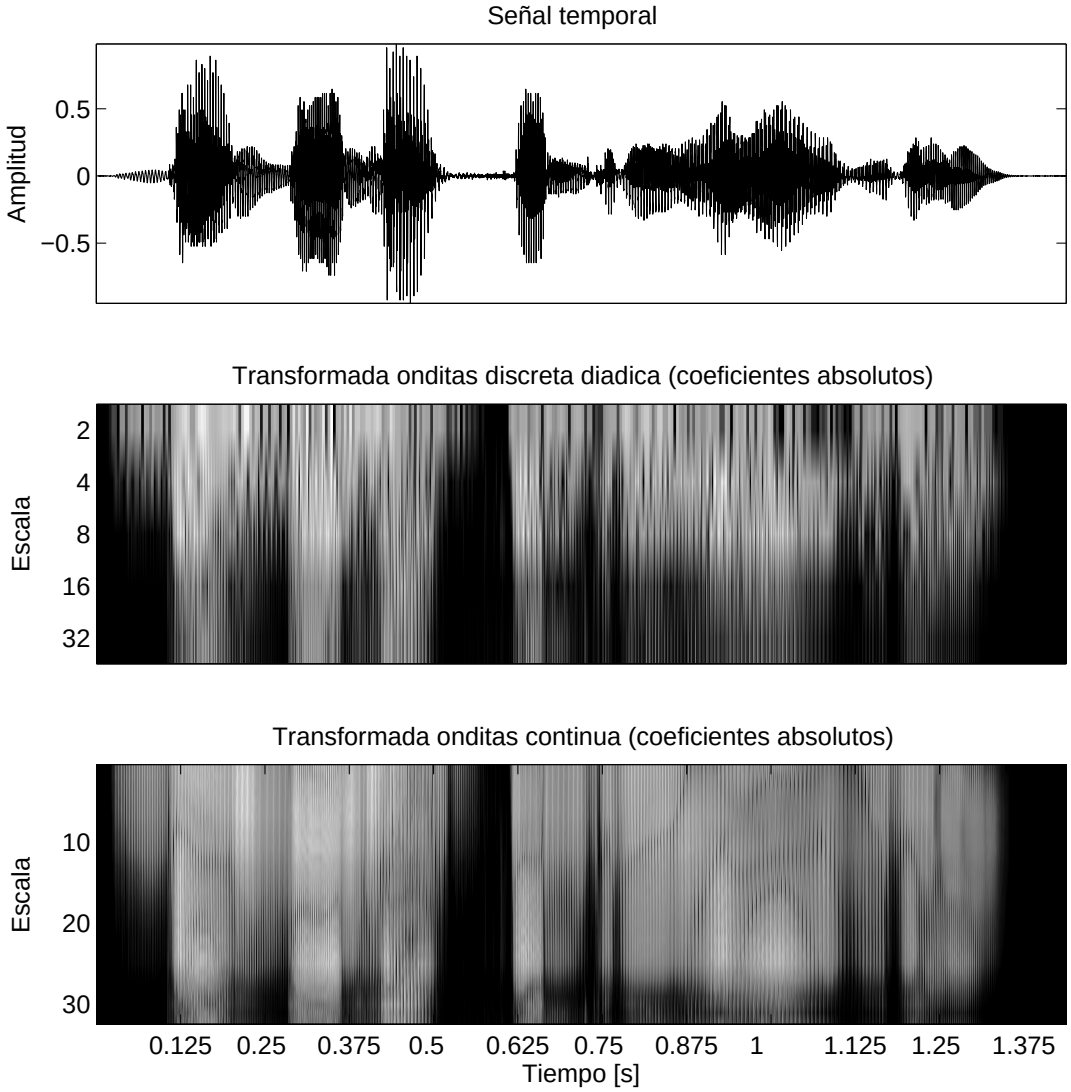


Figura 4.3. Señal temporal, CWT y DWT de la frase "¿Dónde nace el río Ebro?", perteneciente al corpus Albayzin [Moreno et al., 1993].

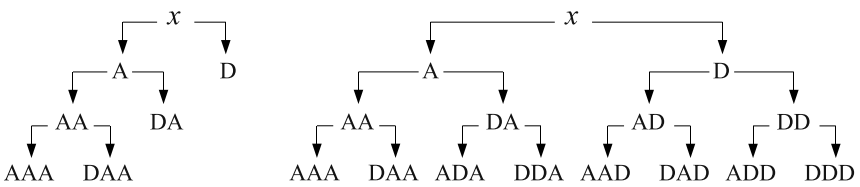


Figura 4.4. Árboles de descomposición de una señal x: a) DWT, b) WPT completa. Las letras A y D representan un nivel de aproximación y un nivel de detalle respectivamente.

de igual manera que las aproximaciones (componentes de bajas frecuencias). Así, el árbol completo de descomposición de la WPT se obtiene mediante

$$c_{j+1}^{2p}[m] = \sqrt{2} \sum_{n=-\infty}^{\infty} g[n-2m] c_j^p[n], \tag{4.10}$$

$$c_{j+1}^{2p+1}[m] = \sqrt{2} \sum_{n=-\infty}^{\infty} h[n-2m] c_j^p[n], \tag{4.11}$$

donde $g[n]$ y $h[n]$ son los filtros definidos en la Sección 4.2.1, j es la profundidad del nodo y p es el índice de los nodos en el mismo nivel. Cada c_j^p con p par está asociado a las aproximaciones y cada c_j^p con p impar está asociado a los detalles.

Una vez seleccionado el árbol adecuado, el análisis mediante paquetes de onditas permite representar la información en un plano tiempo-escala más flexible. Para la selección de este árbol es posible aprovechar conocimiento *a priori* de las características de una señal y obtener una representación eficiente en el dominio transformado. Por otro lado, la familia de árboles de descomposición mediante la transformada paquete de onditas ofrece una gran cantidad de combinaciones para descomponer una misma señal. Elegir una de éstas posibles combinaciones para una aplicación particular representa un problema interesante, el cual se resuelve comúnmente para el caso ortogonal utilizando diversos criterios para elegir la descomposición más conveniente. Por ejemplo, para el caso de compresión de señales se utilizan criterios basados en la entropía de la energía normalizada, lo

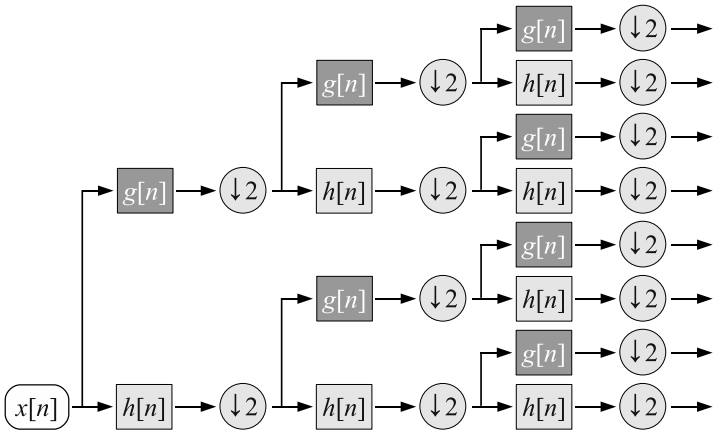


Figura 4.5. Algoritmo de descomposición de la WPT completa de una señal discreta $x[n]$.

que se denomina como *mejor base ortogonal* [Coifman y Wickerhauser, 1992]. Otra posibilidad más cercana al planteo de este trabajo es la denominada *base discriminante local*, que proporciona una base ortogonal adecuada para la clasificación de señales [Saito, 1994].

4.3. Descripción de la optimización

Como se mencionó anteriormente, mediante la WPT se obtiene un conjunto redundante de coeficientes como representación de una señal. En estos coeficientes se evidencian características que son relevantes para la clasificación, pero al mismo tiempo la gran cantidad de los mismos dificulta la tarea del clasificador. Esto se debe a que la cantidad de datos necesarios para el entrenamiento del clasificador crece exponencialmente con la cantidad de dimensiones. Por este motivo resulta necesario seleccionar un conjunto de características que permita discriminar lo mejor posible a las distintas clases de señales, de manera de concentrar en éstas la atención del clasificador.

En los métodos de selección de características denominados *wrappers*, un algoritmo heurístico de búsqueda y optimización envuelve a un algoritmo de clasificación [Hofmann et al., 2004; Kohavi y John, 1997]. Es decir, el algoritmo de selección de características lleva adelante la búsqueda de un conjunto óptimo empleando el clasificador como función objetivo para evaluar cada alternativa. El

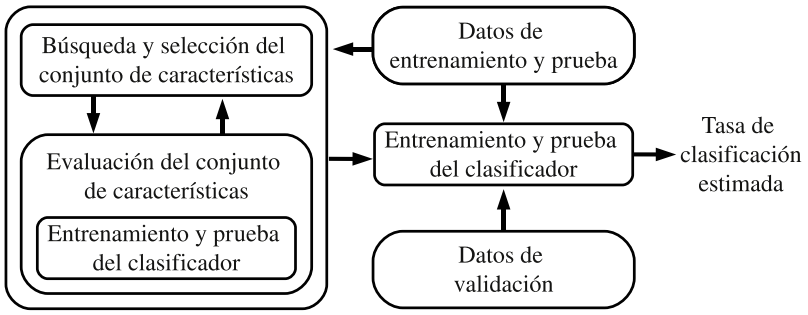


Figura 4.6. Esquema general un algoritmo *wrapper* para la selección de un conjunto de características.

clasificador se utiliza con un conjunto de datos, que usualmente consiste en una partición de entrenamiento y otra de prueba para realizar una validación cruzada [Yu y Cho, 2006]. En cada caso, este conjunto de datos estará representado por el conjunto de características bajo evaluación. Dicho de otra manera, el algoritmo de clasificación es utilizado como una caja negra, y la tasa de acierto es el objetivo que se desea maximizar. Una vez obtenido el conjunto de coeficientes que maximice la clasificación para el conjunto de prueba, el clasificador usualmente es evaluado con un conjunto de datos de validación. En la Figura 4.6 se puede observar el esquema general de un *wrapper*.

En esta propuesta el *wrapper* consiste en un GA que realiza la selección de los coeficientes de la transformada paquetes ondita más relevantes para la clasificación. Para medir la bondad de los distintos conjuntos de características durante la búsqueda este *wrapper* utiliza un clasificador basado en LVQ (Sección 1.3.1). La elección de este tipo de clasificador, en oposición a uno basado en HMM, se debe a su menor costo computacional. La hipótesis subyacente es que facilitando la tarea de este clasificador relativamente sencillo se maximiza la separabilidad de las clases, lo que también favorece al desempeño de un clasificador más complejo como los que se utilizan actualmente en RAH. El esquema del método *wrapper* propuesto se presenta en la Figura 4.7.

4.3.1. Pre-procesamiento

Los tramos de voz se procesan utilizando una implementación de la WPT[1] que aplica los filtros correspondientes, como se explicó en la Sección 4.2.2, su-

[1] La implementación utilizada fue diseñada en base a los algoritmos correspondientes a la DWT incluida en las librerías *Numerical Recipes* [Press et al., 1992a]

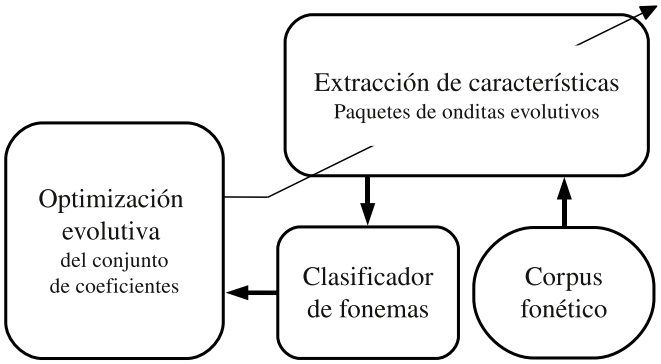


Figura 4.7. Esquema general del *wrapper* para la optimización del conjunto de coeficientes de la WPT.

cesivamente hasta obtener seis niveles de descomposición. Al aplicar la WPT se obtiene, por cada nivel, tantos coeficientes como muestras se tomen de la señal original. Teniendo en cuenta que se trabajó con señales de 256 muestras, esto significa que en último nivel del árbol se tienen 64 nodos de 4 muestras cada uno. Lo que implica que el árbol completo de WPT, incluyendo la señal original (o nodo raíz), tiene 127 nodos y 1792 coeficientes (Figura 4.8).

Para reducir las dimensiones del espacio de búsqueda se realiza una integración por bandas de los coeficientes de la transformada. Esta operación consiste formar grupos de coeficientes dentro de cada banda y sumar sus cuadrados. De esta forma, resulta un coeficiente de energía por cada grupo que se defina. En la Tabla 4.1 se pueden apreciar los detalles del esquema de integración utilizado. La Figura 4.9 ilustra el esquema de integración para la primera mitad del árbol de descomposición, desde el nivel 1 hasta el 6, mientras que la otra mitad se integra de la misma manera. Cada uno de los cuadrados pequeños representa un coeficiente de la descomposición WPT, mientras que las zonas grises y blancas delimitan los diferentes grupos de integración. Las líneas gruesas separan a los distintos nodos del árbol en cada nivel de descomposición. Este esquema de integración ha sido diseñado heurísticamente, considerando las bandas frecuenciales más importantes en el habla.

Los valores resultantes de esta integración por bandas son luego normalizados de forma tal que todos los coeficientes de energía de un patrón dado estén entre 0 y 1. Es decir, si $w_p[k]$ es uno de los coeficientes de energía del patrón p,

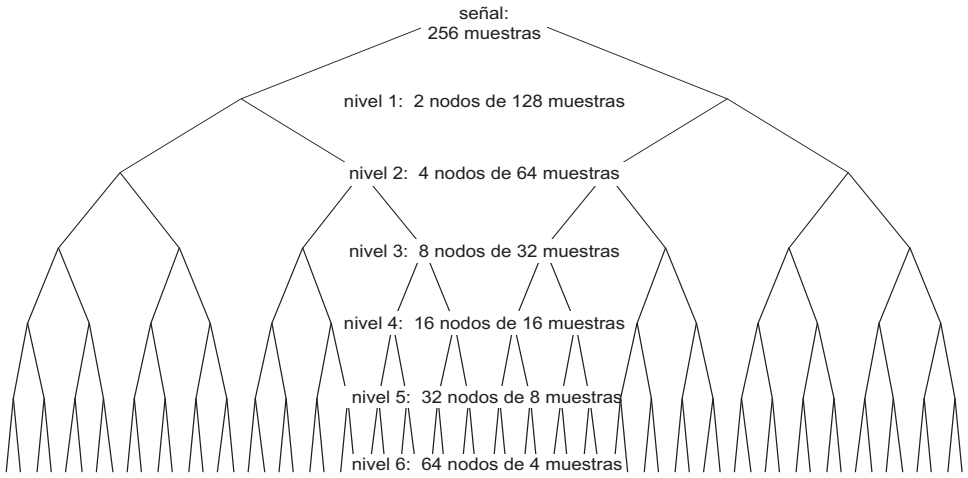


Figura 4.8. Árbol de descomposición obtenido mediante la WPT.

entonces el coeficiente normalizado según este criterio será:

$$\hat{w}_p[k] = \frac{w_p[k]}{\max\limits_{\forall i}\{w_i[k]\}}. \tag{4.12}$$

Una vez obtenidos los valores normalizados de los coeficientes de la transformada se procede a crear los archivos de entrenamiento y prueba para utilizar con el clasificador.

4.3.2. Algoritmo de optimización

El método propuesto consiste en un modelo de evolución simple con cromosomas binarios. Más precisamente, se trata de un algoritmo genético en el cual cada individuo representa una combinación diferente de los elementos provenientes de la integración por bandas de los coeficientes de la WPT.

Cada gen que compone un cromosoma representa uno de los 208 coeficientes de la integración. Si el alelo es 0 indica que este elemento no debe tenerse en cuenta para representar una señal. Por el contrario, un valor 1 indica que sí debe tenerse en cuenta. Para la inicialización se completa cada una de las posiciones de cada cromosoma de la población con valores 0 o 1 elegidos aleatoriamente. La

Nivel	Nodos	Muestras por nodo	Grupos por nodo	Muestras por grupo	Grupos por nivel
1	2	128	8	16	16
2	4	64	8	8	32
3	8	32	4	8	32
4	16	16	2	8	32
5	32	8	1	8	32
6	64	4	1	4	64
				Total de coeficientes de energía	208

Tabla 4.1. Esquema de integración aplicado al árbol de paquete de onditas para una señal de 256 muestras.

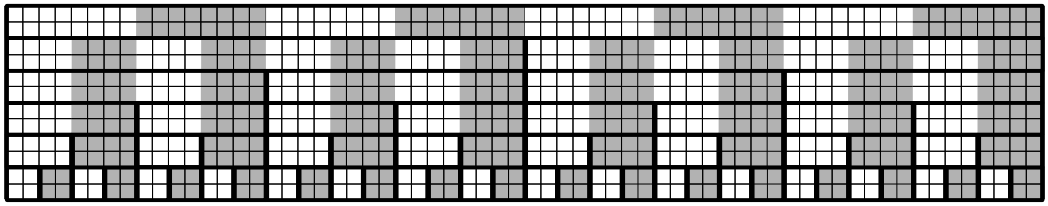


Figura 4.9. Esquema de integración por bandas de frecuencia (mitad del árbol).

representación puede ser redundante, por lo que no existen restricciones respecto a qué elementos existen. La Figura 4.10 ilustra un ejemplo de esta codificación con un cromosoma de 80 genes y un árbol WPT integrado en un total de 80 coeficientes distribuidos en 3 niveles.

Los individuos se deben escoger de manera que las combinaciones de coeficientes que representen permitan separar lo mejor posible las diferentes clases de patrones correspondientes a distintos fonemas. Con este fin se propone la utilización del método LVQ optimizado como función objetivo, tomando como valor de aptitud el resultado de la clasificación. Para acelerar el entrenamiento del clasificador, los prototipos se inicializan lo más cerca posible del centro de la distribución de la clase a la que pertenecen. Para ésto, se evalúa el conjunto de prototipos en base a los patrones de entrenamiento (en un orden aleatorio) según la regla del vecino más cercano.

El entrenamiento del clasificador, como se explicó en la Sección 1.3.1, consiste en modificar el conjunto de vectores prototipos de manera que al finalizar el proceso éstos aproximen lo mejor posible la distribución de las distintas clases

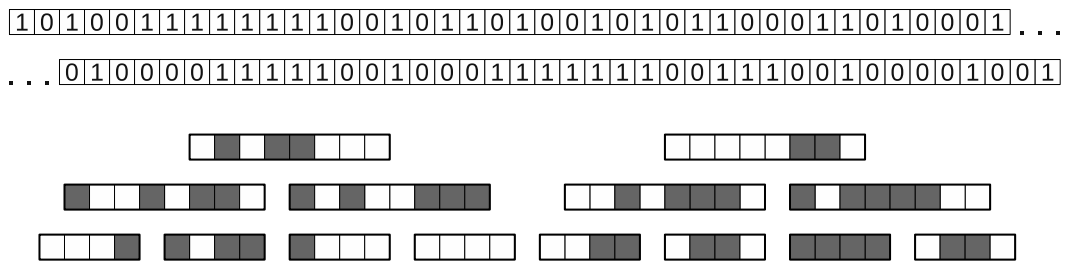


Figura 4.10. Ejemplo de codificación con un cromosoma de 80 genes y el árbol de la WPT correspondiente. Los rectángulos representan los nodos del árbol, los cuadros blancos los coeficientes utilizados y los negros los coeficientes no utilizados.

de patrones. Una vez finalizado el entrenamiento se realiza la prueba para obtener un resultado de clasificación. Es decir, para cada patrón del conjunto de prueba se busca el prototipo más cercano a éste. Si el patrón y el prototipo más cercano pertenecen a la misma clase se lo cuenta como una clasificación correcta y si pertenecen a clases diferentes se lo cuenta como una clasificación incorrecta. Para esto se particionan los datos en tres grupos, uno para entrenamiento, uno para prueba y otro diferente para validación. La particiones de entrenamiento y prueba son utilizadas durante la evolución del GA para adaptar los prototipos del clasificador y estimar la capacidad de generalización respectivamente. Es decir, a partir de clasificar los patrones de prueba mediante la cuantización lograda con la partición de entrenamiento se obtiene el porcentaje de clasificación utilizado como valor de aptitud. La partición de validación es utilizada al finalizar la evolución para comprobar la capacidad de generalización de la mejor solución encontrada. En este último caso, se entrena el clasificador con las particiones de entrenamiento y prueba, y se realiza la clasificación de los patrones de validación.

Existen infinidades de configuraciones posibles para entrenar el clasificador, es decir, variando cada uno de los parámetros se puede obtener resultados significativamente diferentes. Para encontrar una configuración adecuada, teniendo en cuenta el desempeño y el costo computacional, se realizaron numerosas pruebas variando los diferentes parámetros. En estas pruebas se emplearon sólo los patrones de entrenamiento y prueba para que los datos de validación no influyan de ninguna manera en el aprendizaje.

La separación entre los datos de entrenamiento, prueba y validación se realiza una única vez antes de iniciar el GA, de manera que se utiliza la misma para evaluar a todos los individuos. En los archivos correspondientes se almacenan,

para cada patrón, los 208 valores provenientes de la integración por bandas de los coeficientes de la WPT. Durante la evolución, en el momento de evaluar un individuo determinado, la rutina que realiza el cálculo del valor de aptitud extrae de estos archivos únicamente los valores indicados por el cromosoma del correspondiente individuo.

4.4. Corpus de habla y configuración

4.4.1. Descripción del corpus de habla

Para realizar la experimentación se utilizó un subconjunto del corpus geográfico Albayzin [Moreno et al., 1993], denominado Minigeo. Este corpus geográfico está formado por frases correspondientes a una tarea de consulta a una base de datos geográfica. El subconjunto utilizado está formado por 600 archivos de audio, y cada uno de estos archivos almacena una emisión de 12 hablantes diferentes (6 varones y 6 mujeres) de entre 15 y 55 años de edad. El corpus de habla está segmentado fonéticamente, por lo que para cada frase registrada se dispone de la información de las muestras en que comienza y finaliza cada fonema en el registro de voz, con su correspondiente etiqueta.

En las pruebas se incluyeron los fonemas `/a/`, `/e/`, `/i/`, `/o/`, `/u/`, `/b/`, `/d/`, `/p/` y `/t/`. Las cinco vocales fueron incluidas por su obvia importancia en nuestro idioma, mientas que los últimos cuatro fonemas pertenecen a la clase de los oclusivos [Quilis, 1993] y se eligieron porque sus similares características los hacen particularmente difíciles de clasificar.

4.4.2. Configuración del algoritmo

Se realizaron numerosos experimentos con el objetivo de encontrar una representación más apropiada para la clasificación de las señales de habla y en la sección siguiente se detallan los que arrojaron mejores resultados. En cada uno de estos experimentos no sólo se han utilizado diferentes parámetros sino también se probaron estrategias tendientes a mejorar la evolución.

Para la elección de la familia de onditas no existen criterios claros que destaquen una frente a otras en una aplicación particular [Rufiner, 1996]. Por esta razón se decidió experimentar con las más difundidas, entre las cuales podemos citar las de Meyer, Daubechies, Symmlets, Coiflets y Splines [Daubechies, 1992].

En base a las pruebas realizadas con las diferentes familias se optó por utilizar la ondita Coiflets de orden 4.

En el algoritmo genético se utilizó el método de la ruleta para la selección de los individuos en cada generación y se empleó la estrategia de reemplazo con elitismo en la cual se conserva el mejor individuo para la generación siguiente. Se implementaron las operaciones genéticas de mutación y cruza simple.

4.5. Resultados y discusión

En esta sección se presentan y analizan los resultados obtenidos en la optimización de los coeficientes de la WPT. En primer lugar se detallan los experimentos que fueron realizados considerando solamente las vocales del español en la clasificación. Luego se exhiben los resultados obtenidos en la optimización de la representación para la clasificación de nueve fonemas del mismo idioma. Luego se analizan y comparan estos resultados con aquellos alcanzados empleando las representaciones presentadas en el capítulo anterior.

4.5.1. Clasificación de las vocales

El primer experimento se realizó sólo con las cinco vocales del español. En este caso se utilizó un conjunto de referencia (codebook) de 50 vectores, es decir, de aproximadamente 10 prototipos para cada fonema. Para el entrenamiento del clasificador en la evaluación de cada individuo se realizaron 8 épocas de entrenamiento empleando 415 patrones y una validación cruzada con otros 45 patrones. La velocidad de aprendizaje inicial para cada prototipo se ajustó en 0,05 habiendo comprobado que este valor permite la estabilización de los vectores del conjunto de referencia. En la Tabla 4.2 se pueden apreciar los porcentajes de acierto para cada vocal. Como se puede observar, se obtiene un resultado mucho mejor en comparación con la DWT. Estos primeros resultados fueron reportados en [Vignolo y Milone, 2005a,b].

4.5.2. Clasificación de nueve fonemas

En el segundo caso se consideraron, además de las cinco vocales, los fonemas /b/, /d/, /p/ y /t/. Se utilizó un codebook de 117 vectores (13 prototipos por fonema) y la tasa de aprendizaje inicial para cada uno de estos se ajustó en 0,02, habiendo comprobado que este valor permite la estabilización de los vectores del

Fonema	DWT	POE
/a/	28,00	**88,89**
/e/	36,00	**55,56**
/i/	**92,00**	88,89
/o/	76,00	**77,78**
/u/	38,00	**100,0**
Total	54,00	**82,22**

Tabla 4.2. Porcentajes de acierto para el primer experimento realizado.

conjunto de referencia. Para el entrenamiento del clasificador se realizaron sólo 6 épocas empleando 1449 patrones y para efectuar el prueba se utilizaron 252 patrones. El tamaño de la población fue de 100 individuos y las probabilidades de cruza y mutación fueron de 0,9 y 0,05 respectivamente. La mejor solución, que fue encontrada luego de 26 generaciones permitió obtener un reconocimiento total de 57,94 % [Vignolo y Milone, 2006].

Como se mencionó anteriormente, para obtener un porcentaje de clasificación con cada individuo se utiliza una partición de prueba durante la evolución. Dado que la inicialización del vector de prototipos del clasificador se realiza de manera aleatoria es razonable que el porcentaje obtenido varíe de una realización a otra. Es por este motivo que para obtener una mejor estimación de los porcentajes de reconocimiento se repitió diez veces el entrenamiento y la prueba del clasificador con una partición de datos diferente, utilizando el cromosoma del mejor individuo. Dicha partición consiste en 2637 patrones de entrenamiento y otros 450 patrones de prueba o validación, los cuales no fueron utilizados durante la evolución. De esta manera se obtuvo como promedio un 53,69 % de clasificaciones correctas, con un desvío estándar de 2,33 %.

Como se puede apreciar, en la validación se obtuvo un porcentaje de aciertos considerablemente inferior al obtenido en la evolución. Este problema se puede atribuir a que los individuos se adaptan a los conjuntos de entrenamiento y prueba utilizados durante la evolución. Es decir, que la búsqueda converge a un conjunto de coeficientes que favorece la clasificación de los patrones utilizados en la prueba y no para cualquier conjunto de patrones en general. Otro factor que puede influir en este problema es la inicialización de los prototipos, que posee cierta aleatoriedad. Esto afecta en cierto grado el resultado de clasificación en la evaluación de

cada individuo y consecuentemente a la evolución.

Aumento del tamaño del conjunto de prototipos. En el siguiente experimento se aumentó el tamaño del conjunto de prototipos a 144 con el objetivo de estudiar cómo reducir el efecto de la inicialización. En este caso el algoritmo genético convergió en 275 generaciones y el mejor individuo encontrado permitió obtener un 59,13 % de clasificaciones correctas. Se realizó la validación y al igual que en el caso anterior se obtuvo en promedio un porcentaje de aciertos menor, 55,27 % con un desvío estándar de 2,20 %.

Utilización de una partición de datos diferente. En el caso siguiente se utilizó, durante la evolución, una partición de datos diferente para evaluar el efecto del entrenamiento en los resultados. Se destinaron 1953 patrones para el entrenamiento y 270 para la prueba, los demás parámetros se fijaron igual que en el primer experimento analizado. En esta oportunidad el algoritmo genético encontró el mejor individuo en sólo 24 generaciones y el porcentaje de aciertos obtenido con el mismo fue de 61,48 %. Al igual que en los casos anteriores se realizó la validación entrenando del clasificador diez veces obteniendo en promedio un 57,84 % de aciertos con un desvío estándar de 2,55 %.

Eliminación de la aleatoriedad en la inicialización de los prototipos. El siguiente experimento consistió en utilizar la misma partición de datos y la misma configuración de parámetros pero eliminando la aleatoriedad de la inicialización del conjunto de prototipos. La hipótesis planteada consiste en que si todos los individuos son evaluados exactamente de la misma manera, el algoritmo genético será capaz de encontrar una mejor solución. Luego de las primeras 216 generaciones el algoritmo genético encontró el individuo que permitió obtener un 57,78 % de aciertos.

En este caso el mismo procedimiento de validación dio como resultado un promedio de 56,67 % con un desvío estándar de 2,90 %. De esta manera se logró que la validación arroje un resultado mucho más cercano al obtenido mediante la prueba, sin embargo el desvío pone en evidencia un comportamiento no del todo estable (aunque aceptable).

Inicialización de prototipos fija por generación. Si bien se obtuvo una mejora utilizando la estrategia mencionada anteriormente, se puede pensar que

la evolución fue realizada en busca de optimizar la clasificación basada en una inicialización fija para el algoritmo LVQ. Es decir, que al no aleatorizar la inicialización del clasificador los individuos se adaptan a la misma y la solución encontrada posiblemente sea buena únicamente cuando se utiliza la misma inicialización. En este sentido se planteó la estrategia de cambiar inicialización por cada generación. Esto significa que para evaluar a todos los individuos de una misma generación se inicializan los prototipos del clasificador seleccionando los patrones de entrenamiento en un mismo orden. Luego, al pasar a la generación siguiente, este orden se cambia permitiendo que el algoritmo genético realice la búsqueda de una manera más generalizada. En este experimento el mejor individuo encontrado permitió obtener un resultado de 67,78 % de aciertos, mientras que en la validación se obtuvo un promedio de 53,33 % con un desvío de 2,80 %. Aquí vemos que vuelve a abrirse una brecha significativa entre los resultados de prueba y los resultados de validación.

Utilización de brecha generacional. Por último, se repitió el mismo experimento utilizando una brecha generacional de 10 individuos, además de la estrategia de elitismo, con el objetivo de conservar mejor la información de todas las generaciones para que el mejor individuo encontrado no dependa de una inicialización particular. En este caso con la mejor solución encontrada se obtuvo el 64,07 % de aciertos y el algoritmo genético convergió en 355 generaciones.

Mediante el procedimiento de validación se obtuvo nuevamente un porcentaje de aciertos considerablemente menor. El promedio fue de 59,16 % con un desvío estándar de 2,91 %. Si bien no se ha logrado eliminar la variación de los resultados en las distintas validaciones, los resultados de clasificación obtenidos en los experimentos son satisfactorios y demuestran que los algoritmos implementados permiten encontrar una buena representación para la clasificación de los fonemas considerados [Vignolo et al., 2006a,b].

La Tabla 4.3 resume los resultados de todos los experimentos descriptos en esta sección. Como se puede apreciar, a partir de la configuración inicial y mediante las distintas alternativas consideradas se ha podido mejorar el resultado de validación. Para el caso de la mejor representación obtenida (correspondiente a la última fila de la Tabla 4.3), en la Tabla 4.4 se pueden apreciar los porcentajes de acierto y equivocación para cada fonema. En este caso se obtuvo un 59,16 % de aciertos, y como se puede apreciar, el fonema `/t/` en la mayoría de los casos fue clasificado como `/p/`. Este error es razonable porque los fonemas plosivos tienen sus características espectrales más distintivas en el comienzo (la plosión) y en

Experimento	Mejor aptitud	Generaciones	Validación	Desvío
Configuración inicial	57,94 %	26	53,69 %	2,33 %
Aumento del tamaño del conjunto de prototipos	59,13 %	275	55,27 %	2,20 %
Utilización de una partición de datos diferente	61,48 %	24	57,84 %	2,55 %
Eliminación de la aleatoriedad en la inicialización de los prototipos	57,78 %	216	56,67 %	2,90 %
Inicialización de prototipos constante por generación	67,78 %	151	53,33 %	2,80 %
Utilización de brecha generacional	64,07 %	355	**59,16** %	2,91 %

Tabla 4.3. Resumen de los resultados obtenidos con POE.

los experimentos realizados las muestras se tomaron de la parte central de los mismos. Una mejor solución podría consistir en considerar más de un tramo por cada ocurrencia, de manera de abarcar también las zonas inicio y de transición entre fonemas. Para el caso del fonema `/d/` se tiene un problema similar.

Otro aspecto importante acerca de la optimización es la rápida convergencia del GA. En la Figura 4.11 se puede observar cómo varía la aptitud del mejor individuo, del peor individuo y la aptitud promedio de la población en función del número de generaciones. También se puede observar como varía la aptitud del mejor individuo cuando no se utiliza la estrategia de elitismo. Como se puede ver, en comparación con la estrategia de optimización presentada en el capítulo anterior, en este caso se obtienen resultados comparables en una cantidad mucho más reducida de generaciones.

Debe agregarse que las representaciones obtenidas con este método, es decir los mejores cromosomas, varían en cada experimento y no presentan una estructura simple de analizar a partir de una gráfica. En trabajos futuros se evaluará la posibilidad de obtener conclusiones generales acerca de la estructura de la representación obtenida y en relación con las características de la señal utilizada.

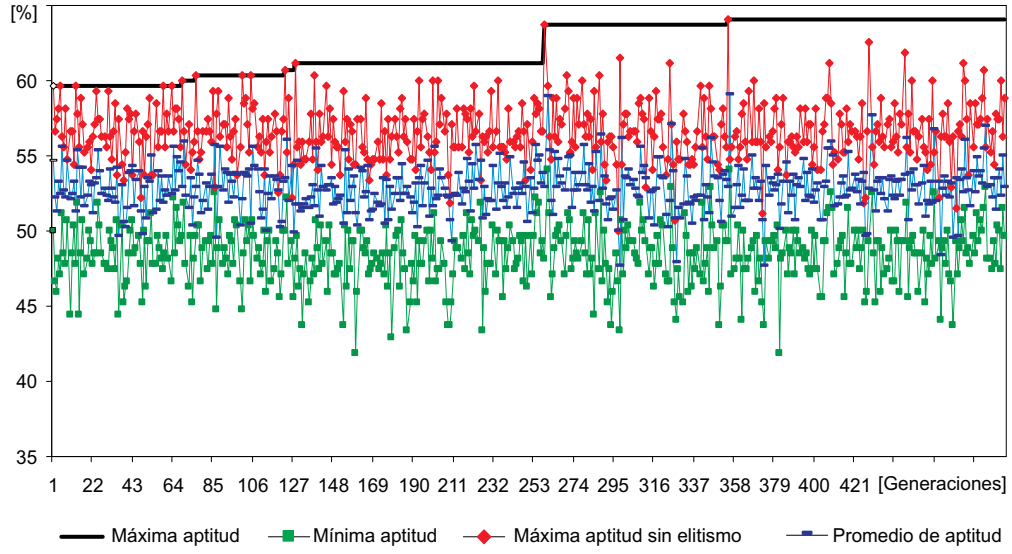


Figura 4.11. Porcentaje de clasificación en función del número de generaciones.

4.6. Análisis comparativo general

4.6.1. Comparación mediante un clasificador LVQ

Para contrastar los resultados, se realizaron pruebas parametrizando las señales con la DWT, y además se representaron las señales mediante los MFCC, HFCC, CCE, CCES y el banco de filtros de Slaney. Los CCE y los CCES empleados en estas pruebas fueron obtenidos mediante bancos de filtros optimizados para los mismos fonemas considerados en la optimización de los POE, de la misma manera que se describió en el Capítulo 3, pero utilizando un clasificador basado en LVQ. Como en este caso las señales están muestreadas con una frecuencia de 8 kHz, los parámetros originales de los bancos de filtros HFCC y Slaney fueron modificados para adecuarlos a este corpus, siguiendo el razonamiento de [Ganchev et al., 2005]. Para el banco de filtros de Slaney se consideraron los primeros 32 filtros, por ser éstos los que se centran antes de la mitad de la frecuencia de muestreo, y se utilizaron los 18 primeros coeficientes cepstrales. Para el caso de HFCC se cubrió el ancho de banda de interés con 24 filtros, se utilizaron los primeros 13 coeficientes cepstrales y, como en los experimentos del Capítulo 3, se utilizó un factor de 5 para determinar el ancho de los filtros. En el caso de

	/a/	/e/	/i/	/o/	/u/	/b/	/d/	/p/	/t/
/a/	**84**,**85**	00,30	00,00	11,82	01,21	01,21	00,30	00,30	00,00
/e/	02,73	**76**,**06**	01,82	05,15	03,64	03,33	01,51	03,64	02,12
/i/	00,00	08,18	**86**,**97**	00,00	00,30	02,73	00,61	00,30	00,91
/o/	25,15	10,61	00,00	**42**,**42**	14,24	04,24	02,73	00,61	00,00
/u/	08,79	00,00	01,51	08,48	**59**,**39**	14,24	00,61	05,15	01,82
/b/	00,30	02,42	00,00	04,54	09,70	**62**,**12**	06,06	13,03	01,82
/d/	10,30	31,82	09,09	07,57	04,54	04,54	**10**,**61**	17,27	04,24
/p/	00,00	00,00	00,00	00,00	00,00	03,03	02,42	**78**,**18**	16,36
/t/	00,00	00,00	00,00	00,30	00,00	04,54	01,82	61,51	**31**,**82**

Tabla 4.4. Matriz de confusión obtenida en la validación del último experimento con POE, para el cual el reconocimiento total fue de 59,16 % (Porcentajes de acierto).

los MFCC también se emplearon los primeros 13 coeficientes, obtenidos mediante un banco de 23 filtros en el rango de 0 a 8 kHz, ya que esta configuración permitió obtener la mayor tasa de aciertos. La Tabla 4.5 presenta dichos resultados, exponiendo los porcentajes de acierto obtenidos con los datos de validación (450 patrones no utilizados durante la optimización). Como se puede apreciar, la parametrización obtenida con el método POE permitió obtener la mejor tasa de clasificación promedio para los nueve fonemas considerados. Esto muestra que mediante la optimización de los coeficientes de la WPT es posible obtener mejoras en los resultados de clasificación, en comparación con las representaciones clásicas.

Además, como se puede observar, los CCES y los CCE también obtuvieron mejoras considerables en comparación con las representaciones utilizadas como referencia. Esto sugiere que las distintas estrategias evolutivas propuestas en esta tesis son útiles para la optimización de representaciones, independientemente del tipo de clasificador utilizado.

Por otro lado, es interesante notar que los resultados muestran que el fonema /d/ representa la mayor dificultad para todas las parametrizaciones, excepto para DWT. También se puede apreciar que el desempeño obtenido con el banco de filtros de Slaney, en relación a los otros, es muy pobre teniendo en cuenta los resultados presentados en el Capítulo 3. Sin embargo, debe tenerse en cuenta que dicho banco de filtros fue diseñado originalmente para emplearse con señales muestreadas a 16 kHz.

Fonema	Bancos de filtros y coeficientes cepstrales					Onditas	
	MFCC	Slaney	HFCC	**CCE**	**CCES**	DWT	**POE**
/a/	82,80	70,40	77,20	81,00	83,40	69,39	**84,85**
/e/	77,40	61,60	62,60	75,80	**87,60**	54,54	76,06
/i/	90,00	61,60	90,00	84,20	**91,80**	84,54	86,97
/o/	46,20	28,40	43,20	38,40	47,60	**54,54**	42,42
/u/	31,60	21,40	34,40	22,40	42,20	31,51	**59,39**
/b/	45,20	39,60	57,80	60,80	54,40	58,48	**62,12**
/d/	08,80	09,00	07,40	15,40	06,20	**59,69**	10,61
/p/	55,60	45,20	39,40	55,80	62,00	04,85	**78,18**
/t/	48,60	**58,40**	54,60	56,20	55,20	31,21	31,82
Promedio	54,02	43,96	51,84	**54,44**	**58,93**	49,86	**59,16**

Tabla 4.5. Comparación de los resultados de reconocimiento de nueve fonemas del español empleando un clasificador basado en LVQ (Porcentajes de acierto).

4.6.2. Evaluación y comparación mediante HMM

Como se mencionó anteriormente, la hipótesis de este trabajo plantea que si mediante la optimización de la representación se logra facilitar la tarea de un clasificador sencillo (basado en LVQ), esto se debe a que la separabilidad de las clases se maximiza, lo que también debería favorecer al desempeño de un clasificador más complejo, como los basados en HMM.

Para comprobar esta hipótesis, se realizó el entrenamiento y la prueba de un clasificador basado en HMM para evaluar cada una de las representaciones de la Tabla 4.5. Al igual que en el Capítulo 3, en este HMM se emplearon mezclas de Gaussianas para modelar las densidades de observación. Se empleó un conjunto de 2484 patrones de entrenamiento y otro de 621 patrones para la prueba del clasificador, ambos con las clases de fonemas balanceadas. Es importante destacar que en la clasificación con HMM se utilizaron todos los tramos sucesivos que conforman a un fonema, a diferencia de la clasificación con LVQ, donde se utilizó solamente el tramo central.

En el caso de la DWT el algoritmo de entrenamiento del HMM no logró converger, lo cual se debe principalmente a que las mezclas de Gaussianas no son capaces de modelar adecuadamente las distribuciones de estos coeficientes [Milone et al., 2010]. Otro factor que influye negativamente para el uso de los coeficientes de la DWT en el entrenamiento de los HMM es la dimensionalidad de la representación resultante. Se decidió, entonces, realizar un post-procesamiento basado en el análisis de componentes principales (PCA, del inglés *Principal Component*

Fonema	Bancos de filtros y coeficientes cepstrales					Onditas	
	MFCC	Slaney	HFCC	**CCE**	**CCES**	DWT+PCA	**POE**
/a/	60,87	60,87	**82,61**	57,97	68,12	50,72	59,42
/e/	65,22	65,22	62,32	**69,57**	65,22	39,13	66,67
/i/	63,77	65,22	**81,16**	72,46	76,81	76,81	76,81
/o/	33,33	28,99	36,23	36,23	27,54	**40,58**	24,64
/u/	44,93	50,72	37,68	42,03	42,03	43,48	**71,01**
/b/	**49,28**	23,19	23,19	28,99	34,78	44,93	30,43
/d/	30,43	18,84	21,74	**39,13**	31,88	21,74	13,04
/p/	30,43	27,54	24,64	46,38	50,72	**53,62**	36,23
/t/	69,57	68,12	42,03	**71,01**	**71,01**	62,32	65,22
Promedio	49,76	45,41	45,73	**51,53**	**52,01**	48,15	49,28

Tabla 4.6. Comparación de los resultados de reconocimiento de nueve fonemas del español empleando un clasificador basado en HMM (Porcentajes de acierto).

Analysis) [Bishop, 2007] para obtener una representación de menor dimensionalidad y cuyos coeficientes tengan una distribución más cercana a una Gaussiana. El mejor resultado para DWT+PCA se obtuvo conservando el 99 % de la varianza, con lo cual se obtuvo una representación de 134 coeficientes.

Si bien POE está basado en la WPT y puede esperarse el mismo problema (en relación a los HMM) que para el caso de la DWT, en este caso no fue necesario aplicar PCA ni otra técnica como post-procesamiento. Por ésto se puede concluir que la integración por bandas de los coeficientes WPT permitió obtener coeficientes con una distribución más adecuada para los HMM, además de reducir la dimensionalidad.

La Tabla 4.6 presenta los resultados de validación con un clasificador basado en HMM, comparando a las representaciones que fueron optimizadas para un clasificador basado en LVQ y las representaciones de referencia. Se puede apreciar que las representaciones optimizadas son las que permitieron obtener los mejores resultados, al igual que en la validación con el clasificador basado en LVQ (Tabla 4.5). Es decir, aunque las representaciones fueron optimizadas empleando un clasificador sencillo, igualmente permitieron obtener buenos resultados mediante HMM. Esto significa que las representaciones optimizadas capturan información relevante para la discriminación, independientemente del clasificador utilizado.

También es interesante notar que la representación obtenida con POE, que permitió obtener el mejor resultado de validación con el clasificador basado en LVQ, tiene un desempeño relativamente menor en el caso del clasificador de HMM. Esto se debe a que, como se mencionó anteriormente, las representaciones basadas en onditas no proveen un conjunto de coeficientes cuyas distribuciones de probabilidad sean adecuadas para estos modelos. Si bien en este caso la integración

por bandas y la técnica de PCA permitieron obtener buenos resultados, queda aún por explorar diferentes alternativas para obtener, a partir de la WPT, un conjunto de coeficientes cuyas densidades puedan ser modeladas adecuadamente mediante mezclas de Gaussianas.

A pesar de las consideraciones precedentes, los resultados de validación presentados en las Tablas 4.5 y 4.6 comprueban que el método propuesto es adecuado para la optimización de una representación basada en onditas. Además, las representaciones obtenidas muestran que mediante este enfoque de optimización evolutiva es posible mejorar el desempeño de la representación clásica utilizada actualmente en el RAH.

4.6.3. Evaluación y comparación en condiciones de ruido

Con el objetivo de evaluar la robustez de la representación optimizada, se añadió ruido a las señales originales. Se realizaron pruebas con diferentes cantidades de ruido, considerando el caso de MMTT, lo que significa que se realizaron las pruebas en diferentes condiciones a las del entrenamiento. Como se explicó en el Capítulo 3, este caso se ajusta más a la realidad que cuando se utilizan las mismas condiciones de ruido en las dos etapas. Cada prueba consistió en un proceso de validación cruzada con diez particiones aleatorias, de 2484 y 621 patrones para entrenamiento y prueba, respectivamente. El proceso de entrenamiento y prueba se repitió para las diez particiones y los resultados fueron promediados. En la Figura 4.12 se pueden observar los resultados promedio, así como las desviaciones estándar estimadas. Estos resultados muestran que, si bien las representaciones optimizadas con la metodología propuesta en el Capítulo 3 (CCE y CCES) ofrecen un mejor desempeño en condiciones de ruido, POE permite obtener buenos resultados en comparación con las otras representaciones en el estado del arte. Más aún, la representación basada en onditas mejora el desempeño de los MFCC en todas las condiciones consideradas, por lo tanto se puede concluir que POE ofrece mayor robustez que la representación clásica. Esto muestra que el enfoque propuesto, para la optimización de representaciones basadas en onditas, representa una alternativa interesante para la extracción de características de las señales de voz.

La Tabla 4.7 muestra información más detallada sobre los desempeños en clasificación obtenidos con MFCC y POE, para el caso de ruido a 40 dB SNR en condiciones MMTT. Cada uno de los porcentajes en estas matrices de confusión es un promedio de los resultados de validación obtenidos con cada una de las diez

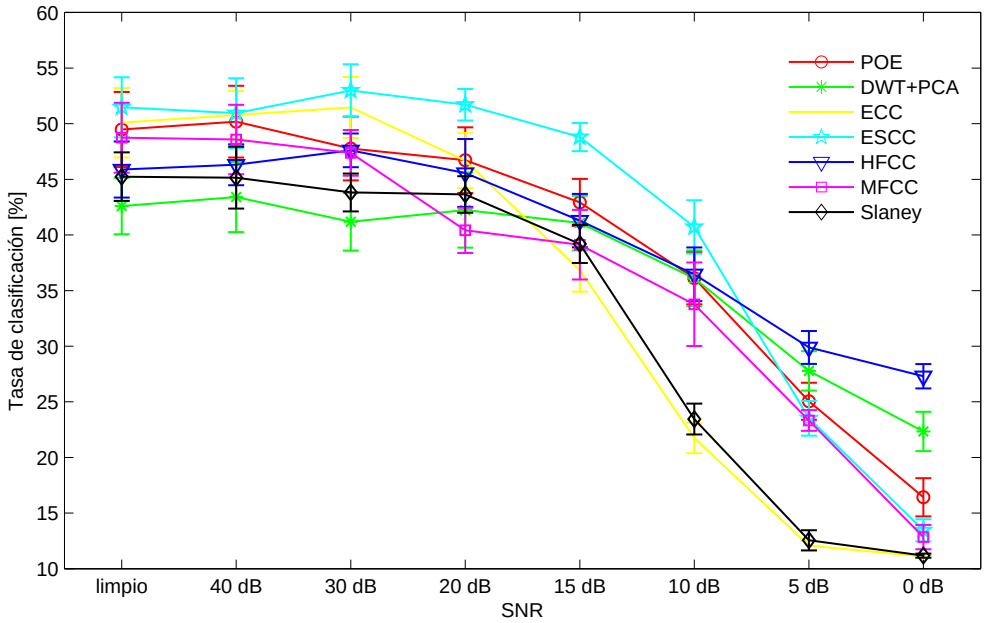


Figura 4.12. Resultados de clasificación con ruido obtenidos con un clasificador basado en HMM.

particiones de datos. Estas matrices de confusión muestran coincidencia entre los fonemas más confundidos para las dos representaciones. Por ejemplo, tanto en el caso de MFCC como en el caso de POE, el fonema /t/ fue confundido en muchos casos con el fonema /p/ y viceversa, lo cual es razonable porque estas consonantes plosivas tienen características espectrales en común. De manera similar, las vocales /o/ y /u/, que son cercanas en el mapa de las formantes, son confundidas entre sí en ambos casos. Similarmente, la Tabla 4.8 muestra las matrices de confusión obtenidas de la clasificación con MFCC y POE en el caso de ruido a 15 dB SNR. Se puede notar que las vocales /a/ y /u/ son frecuentemente mal clasificadas cuando se utiliza MFCC, pero se pueden discriminar significativamente mejor mediante POE. La representación optimizada también introdujo mejoras importantes para la vocal /i/, la cual fue muchas veces clasificada incorrectamente como /d/ cuando se utilizó MFCC. También es interesante notar que los fonemas cuyas tasas de clasificación se vieron más afectadas por el ruido no coinciden para MFCC y POE. Sin embargo, al incrementarse la cantidad de ruido, la cantidad de confusiones entre los fonemas /t/ y /d/ también creció para ambas representaciones. Además, el porcentaje de confusiones entre los fonemas /t/ y /p/

	MFCC									**POE**								
	/a/	/e/	/i/	/o/	/u/	/b/	/d/	/p/	/t/	/a/	/e/	/i/	/o/	/u/	/b/	/d/	/p/	/t/
/a/	***61.7***	08,8	00,2	11,9	02,8	04,5	09,6	00,3	00,3	***57.8***	08,3	00,0	13,2	11,6	05,1	04,1	00,0	00,0
/e/	11,0	***66.4***	06,5	04,8	03,9	03,7	03,8	00,0	00,0	09,1	***61.0***	08,4	04,4	01,2	02,8	11,7	00,3	01,2
/i/	00,2	24,6	***60.9***	02,6	06,0	03,8	02,0	00,0	00,0	00,2	06,8	***77.5***	00,7	03,1	00,6	09,4	00,7	01,0
/o/	15,5	17,1	03,2	***32.3***	13,6	13,3	03,5	00,3	01,2	08,6	10,3	01,2	***35.7***	29,3	05,5	05,7	02,6	01,3
/u/	04,8	04,2	07,4	19,6	***38.0***	15,9	09,0	00,9	00,3	02,3	01,8	02,9	16,1	***57.1***	13,8	04,1	02,1	00,0
/b/	04,7	01,3	03,4	09,9	05,9	***45.2***	18,6	09,9	01,3	04,5	00,3	00,0	11,5	25,8	***29.7***	17,0	06,2	05,1
/d/	06,1	37,0	00,9	03,1	02,8	07,4	***27.8***	05,7	09,4	06,1	23,2	09,4	05,4	07,7	05,4	***26.8***	07,4	08,7
/p/	00,0	00,4	00,0	00,0	01,3	06,0	16,7	***34.1***	41,6	00,0	00,0	00,0	00,2	00,7	03,2	10,0	***52.5***	33,5
/t/	00,0	00,0	02,3	00,2	00,2	02,6	07,0	16,8	***71.0***	00,0	01,0	00,7	00,4	00,9	00,6	12,0	30,7	***53.6***
	Promedio: 48,58 %									Promedio: 50,19 %								

Tabla 4.7. Matrices de confusión obtenidas a partir de las validaciones en condiciones MMTT (40 dB SNR).

	MFCC									**POE**								
	/a/	/e/	/i/	/o/	/u/	/b/	/d/	/p/	/t/	/a/	/e/	/i/	/o/	/u/	/b/	/d/	/p/	/t/
/a/	***04.5***	31,0	00,0	20,1	00,0	03,2	32,8	08,1	00,3	***43.8***	17,3	00,5	18,1	09,8	03,4	07,3	00,0	00,0
/e/	00,0	***80.6***	03,3	00,0	00,0	00,3	15,8	00,0	00,0	04,2	***60.3***	17,7	03,5	01,3	00,9	11,0	00,0	01,2
/i/	00,0	19,1	***53.1***	00,0	00,2	00,2	24,4	00,0	03,2	00,0	11,0	***82.5***	01,2	03,4	00,3	01,7	00,0	00,0
/o/	00,0	25,2	01,8	***26.7***	01,3	09,6	24,6	05,4	05,5	07,1	13,6	00,8	***31.9***	29,6	02,5	12,5	00,2	02,1
/u/	00,0	08,4	11,6	16,8	***03.6***	27,8	29,4	01,3	01,0	02,3	04,4	06,1	19,1	***57.5***	05,2	05,4	00,0	00,0
/b/	00,0	03,7	04,1	03,5	00,4	***12.0***	56,7	08,7	11,0	01,5	09,3	08,6	18,1	29,9	***09.6***	14,9	01,6	06,7
/d/	00,0	29,6	01,2	00,6	00,0	02,8	***49.0***	03,4	13,6	04,5	27,7	12,3	02,1	11,4	00,9	***27.5***	02,2	11,5
/p/	00,0	00,5	00,0	00,0	00,0	00,0	05,1	***48.1***	46,4	00,0	00,5	06,8	01,6	08,7	01,6	19,4	***12.0***	49,4
/t/	00,0	00,0	00,0	00,0	00,0	00,0	13,3	12,2	***74.5***	00,0	01,9	06,7	00,0	03,2	00,7	15,7	10,6	***61.3***
	Promedio: 39,12 %									Promedio: 42,93 %								

Tabla 4.8. Matrices de confusión obtenidas a partir de las validaciones en condiciones MMTT (15 dB SNR).

también se incrementó con el ruido en ambos casos. Otra interesante observación es que, con ambas representaciones, el fonema /t/ es mejor clasificado en el caso de 15 dB SNR que en el de 40 dB SNR. Estos resultados, al igual que los presentados en el Capítulo 3, muestran que el desempeño de clasificación obtenido con la representación clásica en condiciones de ruido puede ser mejorada. Además, las comparaciones realizadas muestran que, mediante la metodología propuesta, se puede proveer de mayor robustez a los sistemas de RAH en el estado del arte, sin incrementar el costo computacional.

4.7. Interpretación de la representación

Para poder realizar un análisis cualitativo de la representación optimizada, se confeccionó un esquema que representa la distribución de los átomos seleccionados en el plano tiempo-frecuencia, en base a los criterios propuestos en [Lewicki,

2002]. El resultado se puede observar en la Figura 4.13, en el cual cada nivel de descomposición se encuentra representado individualmente para facilitar la interpretación. En los gráficos, cada una de las elipses representa un grupo seleccionado a partir del esquema de integración por bandas (Tabla 4.1). Además, el ancho y la localización temporal de cada elipse está determinada por los átomos tiempo-frecuencia que pertenecen al grupo de integración correspondiente (Figura 4.9). Por lo tanto, cada elemento en estas gráficas representa un nuevo átomo tiempo-frecuencia que se obtiene sumando a los átomos onditas originales, de acuerdo al esquema de integración. Esto explica el hecho de que los átomos de los niveles 1 y 2 tienen el mismo ancho, ya que el número de componentes integrados en los grupos del nivel 1 es el doble que en los grupos del nivel 2 (Figura 4.9). La misma explicación es válida para el caso de los átomos de los niveles 5 y 6. Es importante notar que el plano completo de distribución tiempo-frecuencia se obtiene mediante la superposición de las seis gráficas de la Figura 4.13, lo cual resulta en un gran solapamiento entre los átomos de los diferentes niveles. Es oportuno aclarar que al momento de obtener la descomposición de una señal, en realidad, no se integran los átomos tiempo frecuencia sino los coeficientes obtenidos a partir del producto interno de los mismos con la señal. Sin embargo, este análisis es útil para realizar una interpretación de la descomposición obtenida como resultado de la optimización. Una primera observación es que la optimización de la descomposición basada en la WPT resultó en una representación muy redundante y no ortogonal, explotando la redundancia para ganar robustez ante el ruido aditivo. Sin embargo, la representación optimizada utiliza solamente el 50 % de los coeficientes obtenidos a partir de la integración del árbol de descomposición completo. Esto también sugiere que es posible obtener representaciones aún más redundantes y robustas. También es interesante notar que varios de los átomos integrados seleccionados están concentrados en el centro del eje temporal, lo cual puede relacionarse a que los segmentos del corpus fonético fueron extraídos de la parte central de cada pronunciación. También se pueden observar algunos átomos ubicados en los extremos del eje temporal, los cuales pueden relacionarse a los fonemas plosivos.

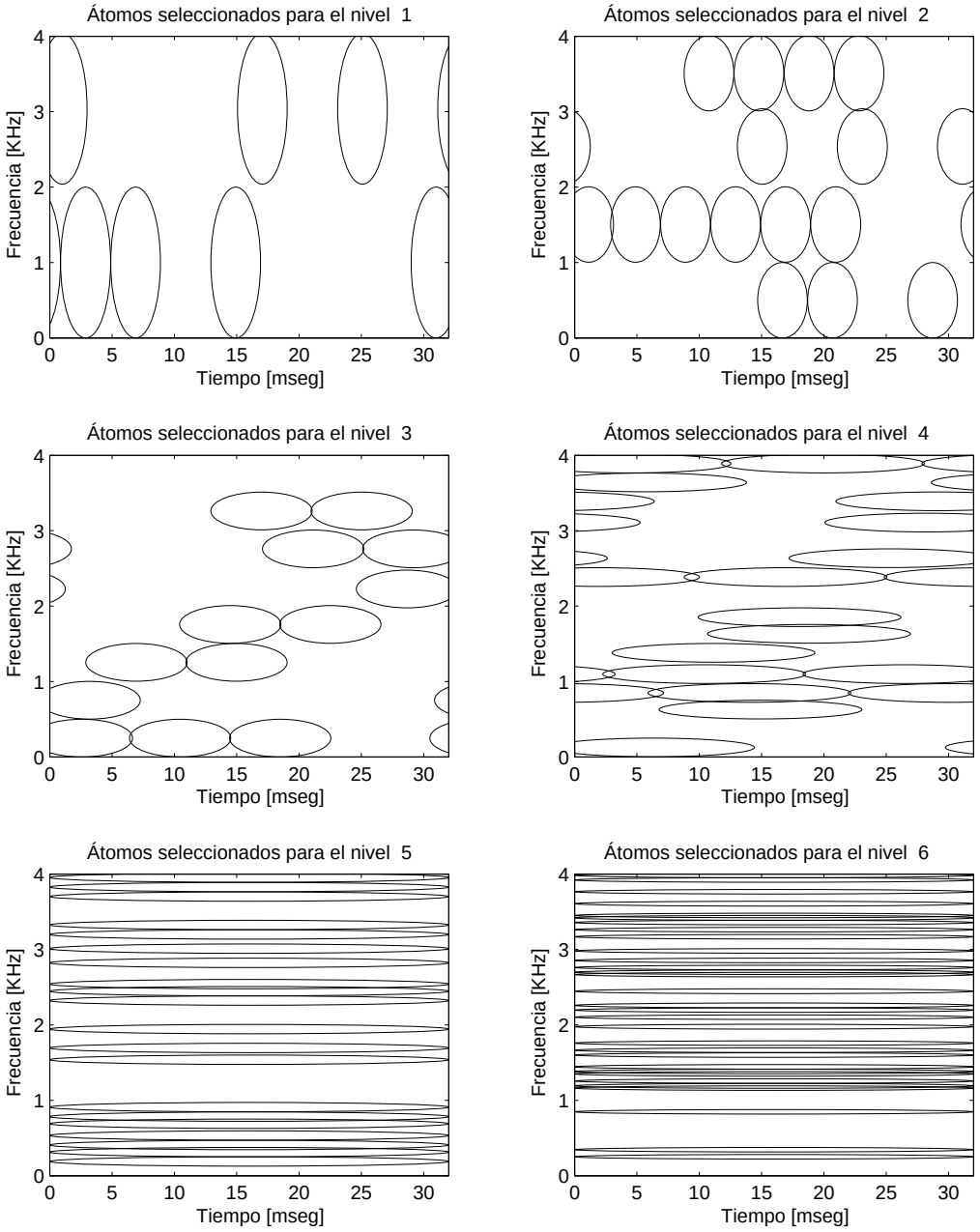


Figura 4.13. Diagrama de cobertura del plano tiempo-frecuencia obtenido para la descomposición optimizada. Para una mejor visualización, cada nivel de descomposición fue representado en un gráfico separado.

Capítulo 5

Conclusiones y trabajos futuros

5.1. Conclusiones

Se han propuesto dos estrategias de optimización que aprovechan las ventajas que proveen las técnicas de computación evolutiva para llevar adelante la búsqueda de una representación que sea más apropiada para la clasificación de las señales de voz. Los resultados obtenidos en la clasificación de distintos grupos de fonemas muestran que las representaciones optimizadas, en comparación con las parametrizaciones clásicas, permiten obtener mejoras significativas. Esto significa que la tarea del clasificador se simplifica debido a que se obtiene una mejor separación de las clases de fonemas. Por lo tanto, estas estrategias proveen representaciones alternativas para las señales de voz, permitiendo mejorar los resultados de las representaciones clásicas en diferentes condiciones de ruido. Por otro lado, utilizando una de estas representaciones en la etapa de procesamiento de un sistema de RAH se puede obtener mayor robustez, sin incrementar el costo computacional y sin modificar la estructura del modelo ni la del algoritmo de entrenamiento.

Coeficientes cepstrales evolutivos

Se ha diseñado un método evolutivo para la optimización de un banco de filtros que constituya parte de una nueva representación para señales de voz. Se introdujo una estrategia de interpolación mediante splines para reducir la cantidad de parámetros en la optimización, proporcionando un espacio de búsqueda más adecuado. Esta nueva codificación permitió reducir significativamente el tamaño de los cromosomas, preservando una amplia variedad de soluciones posibles.

Por otro lado, los operadores de variación específicamente diseñados permitieron explorar adecuadamente el espacio observable.

Los resultados obtenidos en los experimentos muestran que el método propuesto cumple con el objetivo de encontrar una representación robusta para las señales de voz, que permite mejorar el desempeño de un clasificador de fonemas. Dichos resultados también muestran que aún sigue habiendo posibilidad de obtener mejoras sobre el banco de filtros clásico. Por otro lado, en las pruebas realizadas utilizando de una de las representaciones optimizadas en un sistema de reconocimiento continuo, a pesar de que se empleó un conjunto reducido de fonemas en la optimización, se obtuvieron resultados que muestran ciertas mejoras con respecto a la representación clásica.

Además, el uso de un método de selección adaptativa del conjunto de datos para el cálculo de la aptitud de la población permitió evitar la sobre-adaptación de los bancos de filtros sin incrementar el tiempo de procesamiento.

Paquetes de onditas evolutivos

Se ha presentado una técnica que permite seleccionar aquellos componentes, de una descomposición completa mediante la transformada paquete de onditas, que son más relevantes para la clasificación de fonemas. De esta manera se puede optimizar una representación no convencional para alimentar un sistema de reconocimiento automático del habla, como alternativa a los enfoques clásicos.

Los resultados obtenidos en la experimentación muestran que este método cumple con el objetivo de encontrar una representación que permita obtener buenos resultados en la tarea de clasificación de fonemas. Esto se debe a que la representación encontrada permite separar eficientemente las diferentes clases de fonemas, logrando simplificar la tarea del clasificador.

Además, para el conjunto de nueve fonemas del español considerados en los experimentos se puede ver que, en ausencia de ruido, los paquetes de onditas evolutivos proveen una representación con la cual los resultados son mejores incluso en comparación con los CCE. Esto sugiere que el método de optimización evolutiva de la representación basada en paquete de onditas es adecuado para encontrar una representación óptima a partir de un corpus fonético de entrenamiento.

Por otro lado, esta estrategia puede resultar útil para otras aplicaciones relacionadas a las señales de voz, como la identificación de hablantes y la detección de patologías.

5.2. Trabajos futuros

En trabajos futuros se estudiará la posibilidad de utilizar, en los algoritmos evolutivos, operadores de variación que incorporen más información sobre el problema. Además se considerará la utilización de distintos métodos heurísticos de búsqueda y optimización, como el de optimización por enjambre de partículas [Chen y Zhao, 2009; Kennedy y Eberhart, 1995] y búsqueda dispersa [Martí et al., 2006]. Otro aspecto a considerar es la incorporación de medidas de gaussianidad en la función de aptitud, de manera de obtener una representación lo más adecuada posible para realizar la clasificación mediante HMM. Además podría utilizarse un algoritmo evolutivo multi-objetivo [Coello Coello et al., 2007; Salazar et al., 2006], para optimizar dicha medida conjuntamente con el resultado de clasificación y otras características de la representación, como la dimensionalidad y la dependencia estadística de sus coeficientes. Con el objetivo de obtener representaciones que permitan mejorar considerablemente los resultados en un sistema de reconocimiento continuo, se realizaran optimizaciones incluyendo más fonemas en los conjuntos de entrenamiento y prueba utilizados para evaluar cada posible solución. En el mismo sentido, para obtener mayor robustez, también se considerarán distintos tipos y cantidades de ruido en la optimización. Una vez obtenida una representación óptima para la clasificación de fonemas, y considerando que la incorporación en un sistema de RAH no representa modificaciones importantes en el mismo, se realizarán pruebas de reconocimiento continuo en distintas condiciones.

Particularmente, en el caso de los CCE, también se tendrá en cuenta lo siguiente:

- Incorporar de los coeficientes delta y aceleración para añadir información temporal en la representación [Lai et al., 2006], y también incluir parámetros relacionados con estos coeficientes en la optimización.
- Analizar la posibilidad de optimizar conjuntamente otras características de los bancos, como la forma de cada filtro.
- Estudiar la posibilidad de reemplazar el clasificador basado en HMM por otra función objetivo de menor costo computacional. Por ejemplo, una medida de la separabilidad de las clases basada en modelos de mezclas de Gaussianas, relacionada con el clasificador final a utilizar.

Por el lado de los POE se contemplarán, además, las siguientes posibilidades:

- Realizar los ajustes necesarios para la utilización de la mejor representación obtenida en un sistema de reconocimiento automático del habla basado en modelos ocultos de Markov.
- Combinar el algoritmo evolutivo con técnicas numéricas para encontrar una proyección óptima a partir de un diccionario basado en paquetes de onditas.
- Utilizar, e incluso optimizar, diferentes esquemas de integración por bandas.
- Desarrollar una estrategia para optimizar la ondita madre.
- Realizar pruebas de validación con diferentes tipos de ruido.
- Llevar adelante la optimización de una representación basada en paquetes de onditas para los fonemas del inglés, como en el caso de los CCE.

5.3. Publicaciones resultantes del desarrollo de la tesis

- Vignolo, L., Rufiner, H., Milone, D. y Goddard, J., Evolutionary Cepstral Coefficients, en *Applied Soft Computing*, Elsevier Ed., volumen 11, número 4, 2011, `http://dx.doi.org/10.1016/j.asoc.2011.01.012`.
- Vignolo, L., Rufiner, H., Milone, D. y Goddard, J., Evolutionary splines for cepstral filterbank optimization in phoneme classification, en *EURASIP Journal on Advances in Signal Processing, Biologically Inspired Signal Processing: Analysis, Algorithms and Applications*, Hindawi Publishing Corporation, volumen 2011, doi:10.1155/2011/284791 `http://www.hindawi.com/journals/asp/aip.284791.html`.
- Vignolo, L., Rufiner, H., Milone, D. y Goddard, J., Genetic optimization of cepstrum filterbank for phoneme classification, en *Proceedings of the Second International Conference on Bio-inspired Systems and Signal Processing (BIOSIGNALS 2009)*, páginas 179-185, Porto (Portugal), Enero de 2009, INSTICC Press.
- Vignolo, L., Rufiner, H. y Milone, D. Técnicas de computación evolutiva para la extracción de características en reconocimiento del habla, Workshop de Inteligencia Artificial, Tercera Escuela de Posgrado, Red ProTIC, 2008.

- Vignolo, L., Rufiner, H. y Goddard, J., Optimización genética del banco de filtros empleado en el cálculo de coeficientes cepstrales para clasificación de fonemas, *Reporte técnico, SECyT-CONACyT*, Proyecto de Cooperación Bilateral entre Argentina y México, código ME/PA03-EXI/031, "Análisis y reconocimiento robusto del habla mediante técnicas no convencionales", 2007.

- Vignolo, L. y Milone, D., Optimización de paquetes de onditas para la clasificación de señales, IEEE Revista Argentina de Trabajos Estudiantiles (IEEE-RATE), 2(1):29-34, 2006.

- Vignolo, L., Milone, D. y Rufiner, H., Optimización en paralelo de diccionarios de onditas para clasificación de señales, en *10mo Encuentro de Jóvenes Investigadores de la UNL, 1er Encuentro de Jóvenes Investigadores de universidades de Santa Fe*, 2006.

- Vignolo, L., Milone, D., Rufiner, H. y Albornoz, E., Parallel implementation for wavelet dictionary optimization applied to pattern recognition, en *Proceedings of the 7th Argentine Symposium on Computing Technology*, Mendoza, Argentina, 2006.

- Vignolo, L. y Milone, D., Optimización de paquetes de onditas para la clasificación de señales, en *Anales de la XI Reunión de Trabajo en Procesamiento de la Información y Control*, páginas 57-62, Río Cuarto (Córdoba), 2005.

- Vignolo, L. y Milone, D., Procesamiento basado en paquetes de onditas y algoritmos genéticos para clasificación de señales, en *9no Encuentro de Jóvenes Investigadores de la Universidad Nacional del Litoral*, 2005.

Apéndice A

Implementación en paralelo de los algoritmos evolutivos

A.1. Cálculo distribuido

El cálculo paralelo es el uso de múltiples computadoras o procesadores trabajando en conjunto sobre una misma tarea. Cada procesador trabaja en una sección del problema y puede intercambiar información con los demás procesadores.

Según la forma en que el procesador accede a memoria, las computadoras paralelas se pueden clasificar en: computadoras de memoria compartida y computadoras de memoria local. En las computadoras de memoria compartida cada procesador accede a cualquier variable almacenada en la memoria principal, éste es el caso de computadoras de granulado grueso como CRAY 2, Cray YMP, SGI Origin 2000, etc. En las computadoras de memoria local, en cambio, cada procesador accede únicamente a su propia memoria. Este caso es típico de memorias de granulado fino como los hipercubos y los *clusters*.

Un cluster de computadoras es básicamente un conjunto de computadoras desacopladas que cooperan mediante una red de área local de alta velocidad. La configuración más popular de clusters consiste en un conjunto de nodos con sistema operativo Linux[1] y un software de código abierto para la implementación del intercambio de mensajes. Esta configuración es conocida como *Beowulf*[2] y fue diseñada para la ejecución de programas en los cuales se explotan explícitamente las ventajas del paralelismo.

[1] http://www.linux.org/

[2] http://www.beowulf.org/overview/index.html

La manera de programar en computadoras paralelas depende altamente de la manera en que los procesadores acceden a la memoria. El paso de mensajes es un paradigma de programación utilizado ampliamente en computadoras paralelas escalables con memoria distribuida. Aunque existen algunas variantes, el concepto básico de comunicación de procesos mediante mensajes es la base todas ellas.

Una de estas librerías de paso de mensajes es MPI (del inglés *message passing interface*) [Gropp et al., 1999; Pacheco, 1997] cuyo desarrollo comenzó en el año 1992. Su mayor ventaja es la portabilidad a computadoras de cualquier tipo. Esto significa que un mismo programa de paso de mensajes puede ser ejecutado en una variedad de computadoras mientras la librería MPI esté disponible en cada una de ellas. Otra ventaja de MPI es la habilidad de correr de manera transparente en sistemas heterogéneos, es decir, colecciones de procesadores de arquitecturas diferentes. MPI realiza automáticamente cualquier conversión de datos necesaria y utiliza el protocolo de comunicación correcto.

A.2. Paralelización del algoritmo evolutivo

Como se mencionó anteriormente, una característica importante de los AE es que son fácilmente paralelizables. En este caso el AE necesita evaluar cada individuo entrenando y monitoreando un clasificador. Esto implica que para obtener buenos resultados el costo computacional es considerable y así surge la necesidad de ejecutar el algoritmo en computadoras paralelas.

La paralelización del algoritmo fue realizada siguiendo una estrategia maestro-esclavo y se implementó utilizando las librerías MPICH2[3], la cual es una de las implementaciones de MPI más utilizadas. El programa principal o maestro es en sí mismo el AE, es decir se encarga de realizar las operaciones genéticas y la selección de los individuos durante las generaciones. Cada instancia del programa esclavo utiliza las rutinas de clasificación (basadas en LVQ y HMM, según el caso) y se encarga del trabajo más costoso, es decir, el cálculo del valor de aptitud de cada individuo. Esta estrategia es eficiente gracias a que los nodos esclavos no necesitan comunicarse entre ellos y la comunicación con el nodo maestro es mínima. Para evaluar cada individuo la comunicación entre instancias maestro-esclavo consiste únicamente en el cromosoma correspondiente (una secuencia de valores binarios o reales) y el valor de aptitud (un valor real).

Al emplear un cluster de computadoras, se ejecuta el programa maestro en

[3] http://www-unix.mcs.anl.gov/mpi/mpich2/

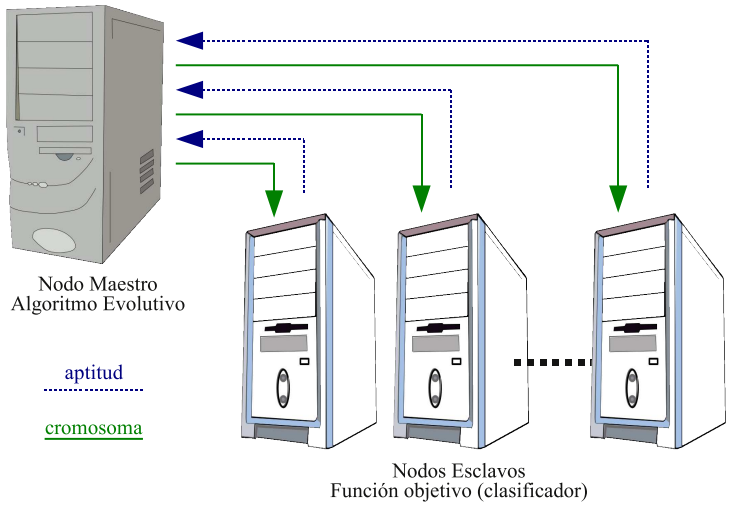


Figura A.1. Estrategia de paralelización del AE.

un nodo y una instancia del programa esclavo en cada uno de los restantes (Figura A.1). De esta manera, el trabajo pesado de calcular el valor de aptitud de cada individuo lo realizan los nodos esclavos, con la ventaja de poder evaluar al mismo tiempo tantos individuos como nodos esclavos se disponga. Durante la evaluación de una población, el programa maestro reparte los individuos entre los nodos esclavos disponibles y envía a cada uno de éstos el cromosoma correspondiente. Cuando un esclavo finaliza el cálculo de aptitud de un individuo envía este valor al programa maestro y recibe un nuevo cromosoma. El proceso se repite hasta completar la evaluación de todos lo individuos de la población.

A.3. Eficiencia de la paralelización

Si bien en las diferentes optimizaciones realizadas se utilizaron once nodos en total, debe tenerse en cuenta que la carga computacional en el nodo maestro no es significativa en comparación con la carga en los nodos esclavos, y por lo tanto en el cálculo de las medidas de eficiencia y aceleración se consideraron diez nodos (o procesadores).

La aceleración (en inglés *speedup*) es una medida de que tanto más rápido

es un algoritmo paralelo que el correspondiente algoritmo secuencial y se define como:

$$S_p = \frac{T_1}{T_p}, \tag{A.1}$$

donde p es el número de procesadores, T_1 es el tiempo de ejecución del algoritmo secuencial (o en un procesador), y T_p es el tiempo de ejecución en paralelo con p procesadores. En este caso la aceleración calculada para uno de los casos de optimización descriptos en los capítulos anteriores es de $S_{10} = 9{,}94$, lo cual hace evidente la importancia de la implementación en paralelo [Vignolo et al., 2006b].

Otra medida útil es la eficiencia, definida como

$$E_p = \frac{S_p}{p}, \tag{A.2}$$

que expresa la relación entre el tiempo real de procesamiento y el tiempo desaprovechado debido a demoras en la comunicación y sincronización. Empleando ambas implementaciones, la secuencial y la paralela, para el mismo experimento, se obtuvo una eficiencia de $E_{10} = 0{,}99$ con diez procesadores. Estas medidas permiten apreciar las importantes ventajas obtenidas con la implementación en paralelo del AE. Más aún, considerando que, estrictamente, la cantidad de procesadores utilizada es once, se obtiene una eficiencia de $E_{11} = 0{,}9$, que también hace evidente las bondades de la paralelización implementada [Vignolo et al., 2006a]. En otros términos, una corrida para la cual se necesitaban 191999 segundos (53,3 horas) con el algoritmo secuencial, se realizó en sólo 19380 segundos (5,4 horas) mediante la paralelización en once nodos.

En el cluster utilizado para la experimentación, los nodos esclavos no poseen unidad de almacenamiento masivo y por lo tanto se utiliza un sistema de archivos de red (NFS, del inglés *network file system*)[4]. Es por ésto que los nodos esclavos deben acceder a los datos de entrenamiento y prueba para el clasificador (función objetivo) desde el nodo maestro, lo cual produce retardos de comunicación. A pesar de ésto, la eficiencia de la implementación en paralelo descripta es cercana a la ideal, lo cual es posible gracias a que la gestión que realiza el NFS permite a los nodos esclavos mantener una copia del corpus fonético en su memoria principal. Como además la comunicación entre las instancias del programa esclavo y el programa maestro es mínima, las demoras debidas a la comunicación son considerables sólo al comienzo de la ejecución.

[4] http://doc.ubuntu-es.org/Network_File_System

Bibliografía

Bäck, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, Oxford, UK. 57

Bäck, T., Hammel, U., y Schewfel, H.-F. (1997). Evolutionary computation: Comments on history and current state. *IEEE Trans. on Evolutionary Computation*, volumen 1, páginas 3–17. 19

Bahl, L. R., Jelinek, F., y Mercer, R. L. (1983). A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volumen PAMI-5, páginas 179 –190. 16

Baker, J. (1974). *Speech Recognition*, capítulo Stochastic Modeling For Automatic Speech Understanding, páginas 521–542. Academic Press, New York. 3

Behroozmand, R. y Almasganj, F. (2007). Optimal selection of wavelet-packet-based features using genetic algorithm in pathological assessment of patients' speech signal with unilateral vocal fold paralysis. *Computers in Biology and Medicine*, volumen 37, páginas 474 – 485. 92

Bishop, C. M. (2007). *Pattern Recognition and Machine Learning*. Springer, 1º edición. 8, 33, 115

Burget, L. y Hermansky, H. (2001). Data Driven Design of Filter Bank for Speech Recognition. En *Text, Speech and Dialogue*, Lecture Notes in Computer Science, páginas 299–304. Springer. 51

Bŏril, H., Fousek, P., y Pollák, P. (2006). Data-driven design of front-end filter bank for lombard speech recognition. En *Proc. of INTERSPEECH 2006 - ICSLP*, páginas 381–384, Pittsburgh, Pennsylvania. 51

Charbuillet, C., Gas, B., Chetouani, M., y Zarader, J. (2007). *Multi Filter Bank Approach for Speaker Verification Based on Genetic Algorithm*, páginas 105–113. Lecture Notes in Computer Science. Springer. 51, 79

Charbuillet, C., Gas, B., Chetouani, M., y Zarader, J. (2009). Optimizing feature complementarity by evolution strategy: Application to automatic speaker verification. *Speech Communication*, volumen 51, páginas 724 – 731. Special issue on non-linear and conventional speech processing. 51, 53

Chen, D. y Zhao, C. (2009). Particle swarm optimization with adaptive population size and its application. *Applied Soft Computing*, volumen 9, páginas 39 – 48. 123

Chipperfield, A. y Fleming, P. (1994). Parallel genetic algorithms: A survey. ACSE Research Report 518, University of Sheffield. 29

Church, K. (1983). Allophonic and phonotactic constraints are useful. En *Int. Joint Conf. on Artificial Intell.*, Karlsruhe, West Germany. 3

Coello Coello, C., Lamont, G., y Van Veldhuizen, D. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation. Springer, Berlin, Heidelberg, 2º edición. 30, 52, 123

Coifman, R. y Wickerhauser, M. V. (1992). Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, volumen 38, páginas 713–718. 100

Corne, D. W., Jerram, N. R., Knowles, J. D., y Oates, M. J. (2001). PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. En *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001*, páginas 283–290. Morgan Kaufmann Publishers. 31

Cortijo, F. y Perez de la Blanca, N. (1997). A compartive study of some non-parametric spectral classifiers. applications to problems with high-overlapping traininig sets. *International Journal of remote Sensing*, volumen 18, páginas 1264–1267. 11

Daubechies, I. (1992). *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA. 106

Davis, G. M. (2002). *Noise reduction in speech applications*. CRC Press. 70

Davis, S. V. y Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE*

Transactions on Acoustics, Speech and Signal Processing, volumen 28, páginas 57–366. 2, 3, 48, 49

de los Cobos Silva, S. G., Goddard Close, J., Gutiérrez Andrade, M. A., y Martinez Licona, A. E. (2010). *Búsqueda y exploración estocástica*. Universidad Autónoma Metropolitana, Iztapalapa, México D.F., 1º edición. 19, 30

Deb, K., Pratap, A., Agarwal, S., y Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, volumen 6, páginas 182 – 197. 31

Deller, J. R., Proakis, J. G., y Hansen, J. H. (1993). *Discrete-Time Processing of Speech Signals*. Macmillan Publishing, NewYork. 35, 37, 45, 49

Demuynck, K., Duchateau, J., Van Compernolle, D., y Wambacq, P. (1998). Improved Feature Decorrelation for HMM-based Speech Recognition. En *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP 98)*, Sydney, Australia. 64, 73, 86

Dias, A. H. y de Vasconcelos, J. A. (2002). Multiobjective genetic algorithms applied to solve optimization problems. *IEEE Transactions on Magnetics*, volumen 38, páginas 1133–1136. 31

Ehrgott, M. (2005). *Multicriteria optimization*. Springer. 30

Eiben, A. E. y Smith, J. E. (2003). *Introduction to Evolutionary Computing*. SpringerVerlag. 56, 57

Ellis, D. P. W. (2005). PLP and RASTA (and MFCC, and inversion) in Matlab. online web resource. 71, 84

Erickson, M., Mayer, A., y Horn, J. (2002). Multi-objective optimal design of groundwater remediation systems: application of the niched Pareto genetic algorithm (NPGA). *Advances in Water Resources*, volumen 25, páginas 51–65. 31

Farooq, O. y Datta, S. (2004). Wavelet based robust sub-band features for phoneme recognition. *Vision, Image and Signal Processing, IEE Proceedings* -, volumen 151, páginas 187 – 193. 92

Ferreira da Silva, A. R. (2001). Evolutionary-based methods for adaptive signal representation. *Signal Processing*, volumen 81, páginas 927–944. 92

Ferreira da Silva, A. R. (2003). Approximations with evolutionary pursuit. *Signal Processing*, volumen 83, páginas 465–481. 92

Fonseca, C. M. y Fleming, P. J. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. En *IEE Colloquium on Genetic Algorithms for Control Systems Engineering*, volumen 6, páginas 1 – 5. 31

Fourier, J. (1888). *Thèorie Analytique de la Chaleur*, volumen I de *Oeuvres de Fourier*. Gathlers-V. Vilars, Paris. 39

Gabor, D. (1946). Theory of communication. part 3: Frequency compression and expansion. *Electrical Engineers - Part III: Radio and Communication Engineering, Journal of the Institution of*, volumen 93, páginas 445 –457.

Ganchev, T., Fakotakis, N., y Kokkinakis, G. (2005). Comparative evaluation of various MFCC implementations on the speaker verification task. En *Proceedings of the SPECOM-2005*, páginas 191–194. 112

Garofalo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallett, D. S., y Dahlgren, N. L. (1993). DARPA TIMIT acoustic phonetic continuous speech corpus CD-ROM. Reporte técnico , U.S. Dept. of Commerce, NIST, Gaithersburg, MD. 63

Gathercole, C. y Ross, P. (1994). Dynamic training subset selection for supervised learning in genetic programming. En *Parallel Problem Solving from Nature – PPSN III*, volumen 866 de *Lecture Notes in Computer Science*, páginas 312–321. Springer. 54, 66

Geva, S. y Sitte, J. (1991). Adaptative nearest neighbor pattern classification. *IEEE Trans. on Neural Networks*, volumen 2, páginas 318–322. 11

Goldberg, D. (1994). Genetic and evolutionary algorithms come of age. En *Communications of the ACM*, volumen 37, páginas 113–119. 22, 24

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional. 21, 22, 23, 24

Gong, Y. (1995). Speech recognition in noisy environments: a survey. *Speech Commun.*, volumen 16, páginas 261–291. 70

Gray, R. (1984). Vector quantization. *IEEE Mag. on Acoustics, Speech and Signal Proc.*, páginas 4–29. 10

Gropp, W., Lusk, E., y Skjellum, A. (1999). *Using MPI*. MIT Press. 128

Haque, S., Togneri, R., y Zaknich, A. (2009). *Auditory Features for Speech Recognition and Enhancement*. VDM Verlag Dr. Müller. 50

Haykin, S. (1994). *Neural Networks. A Comprehensive Foundation*. Macmillan College Publishing Company. 9

Haykin, S. (1999). *Neural Networks*. Prentice Hall, 2º edición. 9

Hermansky, H. (1990). Perceptual linear predictive (plp) analysis of speech. *Journal of the Acoustic Society of America*, volumen 87. 49

Hermansky, H. (1998). Should recognizers have ears? *Speech Communication*, volumen 25, páginas 3 – 27. 2

Hermansky, H. y Morgan, N. (1994). Rasta processing of speech. *Speech and Audio Processing, IEEE Transactions on*, volumen 2, páginas 578 –589. 50

Hess-Nielsen, N. y Wickerhouser, M. V. (1996). Wavelets and Time-Frequency Analisys. *Proceedings of the IEEE*, volumen 84, páginas 523–540. 43, 97

Hofmann, A., Horeis, T., y Sick, B. (2004). Feature selection for intrusion detection: an evolutionary wrapper approach. En *Neural Networks. Proceedings of the 2004 IEEE International Joint Conference on*, volumen 2, páginas 1563 – 1568. 93, 100

Holland, J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press. 26

Hsieh, C.-T., Lai, E., y Wang, Y.-C. (2002). Robust speech features based on wavelet transform with application to speaker identification. *Vision, Image and Signal Processing, IEE Proceedings* -, volumen 149, páginas 108 – 114. 92

Huang, X., Acero, A., y Hon, H.-W. (2001). *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall, Upper Saddle River, NJ, USA. Foreword By-Reddy, Raj. 47, 48, 50

Huang, X. D., Ariki, Y., y Jack, M. A. (1990). *Hidden Markov Models for Speech Recognition*. Edinburgh University Press. 13, 64

Janikow, C. y Michalewicz, Z. (1991). An experimental comparison of binary and floating point representations in genetic algorithms. En *Proc. 4th Int. Conf. Genetic Algorithms*, páginas 31–32. 21, 22

Jankowski, C. R., Vo, H. D., y Lippmann, R. P. (1995). A comparison of signal processing front ends for automatic word recognition. *IEEE Transactions on Speech and Audio Processing*, volumen 4, páginas 251–266. 48

Jelinek, F. (1999). *Statistical Methods for Speech Recognition*. MIT Press, Cambrige, Masachussets. 13, 16, 64

Jones, E., Runkle, P., Dasgupta, N., Couchman, L., y Carin, L. (2001). Genetic algorithm wavelet design for signal classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, volumen 23, páginas 890–895. 92

Kay, S. M. (2006). *Intuitive probability and random processes using MATLAB*. Birkhäuser. 56

Kennedy, J. y Eberhart, R. (1995). Particle swarm optimization. En *IEEE International Conference on Neural Networks*, volumen 4, páginas 1942–1948. 123

Kim, M., Hiroyasu, T., Miki, M., y Watanabe, S. (2004). SPEA2+: Improving the Performance of the Strength Pareto Evolutionary Algorithm 2. En Yao, X., Burke, E., Lozano, J. A., Smith, J., Merelo-Guervós, J. J., Bullinaria, J. A., Rowe, J., Tino, P., Kabán, A., y Schwefel, H.-P., editors, *Parallel Problem Solving from Nature - PPSN VIII*, volumen 3242 de *Lecture Notes in Computer Science*, páginas 742–751. Springer Berlin - Heidelberg. 31

Knowles, J. y Corne, D. (2000). M-PAES: a memetic algorithm for multiobjective optimization. En *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volumen 1, páginas 325 – 332. 31

Kohavi, R. y John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, volumen 97, páginas 273 – 324. 93, 100

Kohonen, T. (1990). Improved versions of learning vector quantization. En *Proc. of the Int. Joint Conf. on Neural Networks*, páginas 545–550, San Diego. 11, 12

Kohonen, T. (1992). New developments of learning vector quantization and the self-organizing map. En *Symposium on Neural Networks; Alliances and Perspectives*, Osaka, Japón. 13

Kohonen, T. (2001). *Self-Organizing Maps*. Springer Series in Information Sciences. Springer, New York, 3º edición. 10, 11

Kohonen, T., Kangas, J., Laaksonen, J., y Torkkola, K. (1996). LVQ Pack: The Learning Vector Quantization Program Package. TR A30, Helsinki University of Technology. 11, 12

Koza, J. (1990). Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems. Reporte Técnico STAN-CS-90-1314, Standford University. 22

Kwon, O.-W. y Lee, T.-W. (2004). Phoneme recognition using ICA-based feature extraction and transformation. *Signal Process.*, volumen 84, páginas 1005–1019. 73

Lai, Y.-P., Siu, M., y B., M. (2006). Joint Optimization of the Frequency-Domain and Time-Domain Transformations in Deriving Generalized Static and Dynamic MFCCs. *Signal Processing Letters, IEEE*, volumen 13, páginas 707–710. 123

Lankhorst, M. M., Van der Laan, M. D., y Halang, W. A. (2003). Wavelet-based signal approximation with genetic algorithms. *Systems Analysis Modelling Simulation*, volumen 43, páginas 1503–1528. 92

Lee, K., Hon, H., y Reddy, R. (1990). An overview of the SPHINX Speech Recognition System. *IEEE Trans. on Acoustics, Speech, and Signal Proc.*, volumen 38. 4

Lewicki, M. S. (2002). Efficient coding of natural sounds. *Nature Neuroscience*, volumen 5, páginas 356–363. 118

Lippmann, R. (1997). Speech recognition by machines and humans. En *Speech Communication*, volumen 22, páginas 1–15. 39

Makhoul, J. (1975a). Linear prediction: A tutorial review. En *Proc. of IEEE*, volumen 63 de *4*, páginas 561–578. 39

Makhoul, J. (1975b). Linear prediction: A tutorial review. *Proceedings of the IEEE*, volumen 63, páginas 561 – 580. 50

Mallat, S. (1989). A theory of multiresolution of signal decomposition: the wavelet representation. *IEEE Trans. Pattern Anal. Machine Intell.*, volumen 11, páginas 674–693. 42, 91

Mallat, S. (1999). *A Wavelet Tour of signal Processing*. Academic Press, 2º edición. 42, 44, 94

Martí, R., Laguna, M., y Glover, F. (2006). Principles of scatter search. *European Journal of Operational Research*, volumen 169, páginas 359 – 372. Feature Cluster on Scatter Search Methods for Optimization. 123

Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag. 21, 22, 26

Milone, D. (2004). Fundamentos del reconocimiento automático del habla. Reporte técnico , Universidad Nacional del Litoral. 16

Milone, D. H. (2003). *Información acentual para el reconocimiento automático del habla*. PhD thesis, Departamento de Electrónica y Tecnología de Computadores, Facultad de Ciencias, Universidad de Granada, España. 17, 38

Milone, D. H., Persia, L. E. D., y Torres, M. E. (2010). Denoising and recognition using hidden markov models with observation distributions modeled by hidden markov trees. *Pattern Recognition*, volumen 43, páginas 1577 – 1589. 114

Moreno, A., Poch, D., Bonafonte, A., Lleida, E., Llisterri, J., Mariño, J., y C. Nadeu (1993). Albayzin speech database, design of the phonetic corpus. Reporte técnico , Universitat Politècnica de Catalunya (UPC). 106

Nasersharif, B. y Akbari, A. (2007). SNR-dependent compression of enhanced Mel sub-band energies for compensation of noise effects on MFCC features. *Pattern Recognition Letters*, volumen 28, páginas 1320 – 1326. Advances on Pattern recognition for speech and audio processing. 51

Nicolson, L. y Cheetham, B. (1993). Simulated annealing applied to the design of IIR digital filters by multiple criterion optimisation. En *Workshop on Natural Algorithms and Signal Proc.*, volumen 6, páginas 1–7. 29

Nilsson, N. J. (1996). Introduction to machine learning. an early draft of a proposed textbook. 8

Oppenheim, A. y Schafer, R. (1989). *Discrete-Time Signal Processing*. Prentice-Hall, Inc., Englewood Cliffs, NJ. 39

Pacheco, P. S. (1997). *Parallel programming with MPI*. Morgan Kaufmann. 128

Press, W., Teukolsky, S., Vetterling, W., y Flannery, B. (1992a). *Numerical Recipes in C, The Art of Scientific Computing*. Cambridge University Press, 2º edición. 101

Press, W. H., Flannery, B. P., Teukolsky, S. A., y Vetterling, W. T. (1992b). *Numerical recipes in C: the art of scientific computing*. Cambridge University Press, 2º edición. 59

Quilis, A. (1993). *Tratado de Fonología y Fonética Españolas*. Biblioteca Románica Hispánica. Editorial Gredos, Madrid. 34, 35, 106

Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, volumen 77, páginas 257 –286. 14

Rabiner, L. y Juang, B. (1986). An introduction to hidden markov models. *ASSP Magazine, IEEE*, volumen 3, páginas 4 – 16. 14, 15

Rabiner, L. y Juang, B.-H. (1993). *Fundamentals of Speech Recognition*. Prentice Hall. 4, 48, 63

Rabiner, L. y Schafer, R. (1978). *Digital Processing of Speech Signals*. Prentice Hall, NJ. 39

Rabiner, L. y Schafer, R. (2011). *Theory and applications of digital speech processing*. Prentice Hall, Upper Saddle River, 1º edición. 4, 5

Rao, S. S. (2009). *Engineering optimization: theory and practice*. John Wiley and Sons. 20

Ravindran, A., Reklaitis, G. V., y Ragsdell, K. M. (2006). *Engineering optimization: methods and applications*. John Wiley & Sons. 20

Ray, S. y Chan, A. (2001). Automatic feature extraction from wavelet coefficients using genetic algorithms. En *Proceedings of the 2001 IEEE Signal Processing Society Workshop*, páginas 233–241. Neural Networks for Signal Processing XI. 92

Reeves, C. R., editor (1993). *Modern heuristic techniques for combinatorial problems*. John Wiley & Sons, Inc., New York, NY, USA. 20

Rufiner, H. (1996). Comparación entre análisis onditas y fourier aplicados al reconocimiento automático del habla. Master's thesis, Universidad Autónoma Metropolitana, Iztapalapa. 106

Rufiner, H. y Goddard, J. (1997). A method of wavelet selection in phoneme recognition. En *Circuits and Systems, 1997. Proceedings of the 40th Midwest Symposium on*, volumen 2, páginas 889 –891 vol.2. 92

Rufiner, H. L. (2005). *Análisis y representación de la voz mediante técnicas no convencionales*. Tesis de doctorado, Facultad de Ingeniería - Universidad de Buenos Aires. 42

Rufiner, H. L. (2009). *Análisis y modelado digital de la voz: Técnicas recientes y aplicaciones*. Universidad Nacional del Litoral, Santa Fe, Argentina, 1º edición. 34, 35, 91

Rutkowski, L. (2005). *Computational Intelligence, Methods and Techniques*. Springer Publishing Company. 9

Saito, N. (1994). *Local feature extraction and its applications using a library of bases*. PhD thesis, Yale University, New Haven, USA. Director-Ronald R. Coifman. 92, 100

Sakoe, H. y Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. on Acoustics, Speech and Signal Proc.*, volumen 26, páginas 43–49. 3

Salazar, D., Rocco, C., y Galván, B. (2006). Optimization of constrained multiple-objective reliability problems using evolutionary algorithms. *Reliability Engineering and System Safety*, volumen 91, páginas 1057–1070. 31, 123

Schell, T. y Uhl, A. (2003). Optimization and assessment of wavelet packet decompositions with evolutionary computation. *EURASIP Journal on Applied Signal Processing*, volumen 2003, páginas 806–813. 92

Skowronski, M. y Harris, J. (2002). Increased MFCC filter bandwidth for noise-robust phoneme recognition. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volumen 1, páginas 801–804. 51

Skowronski, M. y Harris, J. (2003). Improving the filter bank of a classic speech feature extraction algorithm. En *Proceedings of the 2003 International Symposium on Circuits and Systems (ISCAS)*, volumen 4, páginas 281–284. 51

Skowronski, M. y Harris, J. (2004). Exploiting independent filter bandwidth of human factor cepstral coefficients in automatic speech recognition. *The Journal of the Acoustical Society of America*, volumen 116, páginas 1774–1780. 69, 71

Slaney, M. (1998). Auditory Toolbox, Version 2. Technical Report 1998-010, Interval Research Corporation, Apple Computer Inc. 51, 70

So, H., Ching, P., y Chan, Y. (1994). A new algorithm for explicit adaptation of time delay. *IEEE Transactions on Signal Processing*, volumen 42, páginas 1816–1820. 28

Stevens, K.Ñ. (2000). *Acoustic Phonetics*. Mit Press. 35, 64

Tang, K., Man, K. F., Kwong, S., y He, Q. (1996). Genetic algorithms and their applications. *IEEE Signal Processing*, volumen 13, páginas 22–29. 25, 28

Vetterli, M. y Herley, C. (1992). Wavelets and filter banks: Theory and design. *IEEE Trans. Signal Proc.*, volumen 40, páginas 2207–2232. 91

Vignolo, L. y Milone, D. (2005a). Optimización de paquetes de onditas para la clasificación de señales. En *Anales de la XI Reunión de Trabajo en Procesamiento de la Información y Control*, páginas 57–62, Río Cuarto (Córdoba). 93, 107

Vignolo, L. y Milone, D. (2005b). Procesamiento basado en paquetes de onditas y algoritmos genéticos para clasificación de señales. En *9no Encuentro de Jóvenes Investigadores de la Universidad Nacional del Litoral*. 107

Vignolo, L. y Milone, D. (2006). Optimización de paquetes de onditas para la clasificación de señales. *IEEE-RATE: Revista Argentina de Trabajos Estudiantiles*, volumen 1, páginas 29–34. 108

Vignolo, L., Milone, D., y Rufiner, H. (2006a). Optimización en paralelo de diccionarios de onditas para clasificación de señales. En *10mo Encuentro de Jóvenes Investigadores de la UNL, 1er Encuentro de Jóvenes Investigadores de universidades de Santa Fe*. 110, 130

Vignolo, L., Milone, D., Rufiner, H., y Albornoz, E. (2006b). Parallel implementation for wavelet dictionary optimization applied to pattern recognition. En *Proceedings of the 7th Argentine Symposium on Computing Technology*, Mendoza, Argentina. 110, 130

Vignolo, L., Rufiner, H., Milone, D., y Goddard, J. (2009). Genetic optimization of cepstrum filterbank for phoneme classification. En *Proceedings of the Second International Conference on Bio-inspired Systems and Signal Processing (BIOSIGNALS 2009)*, páginas 179–185, Porto (Portugal). INSTICC Press. 52, 57, 67

Vignolo, L., Rufiner, H., Milone, D., y Goddard, J. (2011a). Evolutionary Cepstral Coefficients. *Applied Soft Computing*, volumen 11, páginas 3419 – 3428. DOI: 10.1016/j.asoc.2011.01.012. 54, 72

Vignolo, L., Rufiner, H., Milone, D., y Goddard, J. (2011b). Evolutionary Splines for Cepstral Filterbank Optimization in Phoneme Classification. *EURASIP Journal on Advances in Signal Processing*, volumen Volumen 2011. doi:10.1155/2011/284791. 58, 82, 83, 85

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, volumen 13, páginas 260 – 269. 15

Wang, C., Hou, L. M., y Fang, Y. (2005). Individual dimension gaussian mixture model for speaker identification. En Li, S. Z., Sun, Z., Tan, T., Pankanti, S., Chollet, G., y Zhang, D., editors, *Advances in Biometric Person Authentication*, volumen 3781 de *Lecture Notes in Computer Science*, páginas 172–179. Springer Berlin / Heidelberg. 73

Wickerhauser, M. V. (1991). Lectures on wavelet packet algorithms. En *Lecture notes, INRIA*, páginas 31–99. 91

Widrow, B. y Lehr, M. (1990). 30 years of Adaptative Neural Networks: Perceptron MADALINE and Backpropagation. *Proc. of IEEE*, volumen 78, páginas 1415–1440. 9

Withley, D. (1987). Using reproductive evaluation to improve genetic search and heuristic discovery. En *Proc. 2nd Int. Conf. Genetic Algorithms*, páginas 108–115. 23

Wu, J.-D. y Lin, B.-F. (2009). Speaker identification using discrete wavelet packet transform technique with irregular decomposition. *Expert Systems with Applications*, volumen 36, páginas 3136 – 3143. 91

Wu, Z. y Cao, Z. (2005). Improved MFCC-Based Feature for Robust Speaker Identification. *Tsinghua Science & Technology*, volumen 10, páginas 158 – 161. 51

Yeganeh, H., Ahadi, S., Mirrezaie, S., y Ziaei, A. (2008). Weighting of Mel Subbands Based on SNR/Entropy for Robust ASR. En *Signal Processing and Information Technology, 2008. ISSPIT 2008. IEEE International Symposium on*, páginas 292–296. 51

Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Valtchev, V., y Woodland, P. (2001). *The HTK book, HTK version 3.1*. Cambridge University. 50

Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., y Woodland, P. (2000). *HMM Toolkit*. Cambridge University. 64

Yu, E. y Cho, S. (2006). Ensemble based on GA wrapper feature selection. *Computers & Industrial Engineering*, volumen 51, páginas 111 – 116. Special Issue on Computational Intelligence and Information Technology: Applications to Industrial Engineering, 33rd ICC&IE - Computational Intelligence & Information. 93, 101

Zhang, B.-T. y Cho, D.-Y. (1999). Genetic Programming with Active Data Selection. En *Lecture Notes in Computer Science*, volumen 1585, páginas 146–153. 53

Zhang, B.-T. y Veenker, G. (1991). Focused incremental learning for improved generalization with reduced training sets. En Kohonen, T., editor, *Proc. Int. Conf. Artificial Neural Networks*, volumen 1585, páginas 227–232, North-Holland. 53

Zhou, X., Fu, Y., Liu, M., Hasegawa-Johnson, M., y Huang, T. (2007). Robust Analysis and Weighting on MFCC Components for Speech Recognition and Speaker Identification. En *Multimedia and Expo, 2007 IEEE International Conference on*, páginas 188–191. 51, 70

Esta tesis fue escrita en LaTeX y editada con Kile.