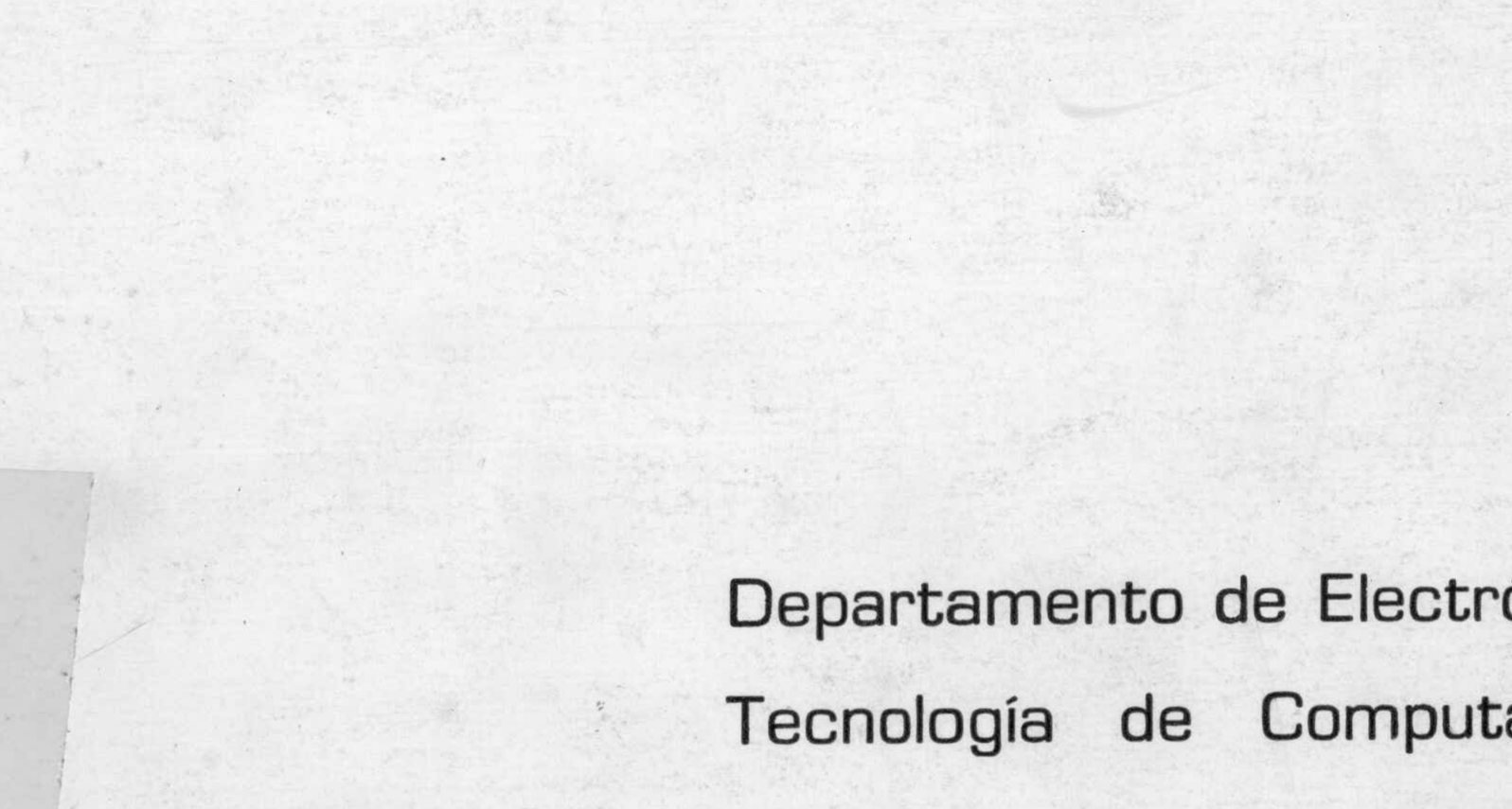

Departamento de Electrónica y
Tecnología de Computadores

FACULTAD DE CIENCIAS
18071-GRANADA

MODELOS DE MARKOV CON CUANTIZACION DEPENDIENTE PARA RECONOCIMIENTO DE VOZ

MODELOS DE MARKOV CON CUANTIZACION DEPENDIENTE PARA RECONOCIMIENTO DE VOZ

TESIS DOCTORAL

José Carlos Segura Luna

Dept. Electrónica y Tecnología de Computadores

Universidad de Granada

Granada, Noviembre 1991

A Juán José y María

Quiero expresar mi agradecimiento a todas aquellas personas que, directa o indirectamente han contribuido a la realización de la presenta Tesis Doctoral, a todos ellos mi más sincera gratitud. En especial al profesor Antonio Rubio, director de ésta Tesis, sin cuya dirección y constante apoyo no podría haber finalizado ésta. A los profesores Antonio Peinado y Jesús Díaz por su inestimable trabajo en la realización de la base de datos. Y en general a todos los compañeros del "Grupo de Investigación en Procesamiento digital de Señales y Comunicaciones" por su incondicional apoyo e inestimable ayuda.

INDICE

CAPITULO 1

INTRODUCCION

1.1. RECONOCIMIENTO AUTOMATICO DEL HABLA

Los orígenes del reconocimiento automático del habla se remontan a la década de los 40, cuando el desarrollo de los primeros espectrógrafos (dispositivos capaces de visualizar la distribución de energías en función de la frecuencia y el tiempo para una señal) permitieron vislumbrar la posibilidad de la construcción de dispositivos automáticos capaces de reconocer la voz humana.

Ya en 1952, Davis, Bidulph y Blashek, de los laboratorios Bell, contruyeron el primer dispositivo automático capaz de discriminar con cierta precisión entre los diez dígitos ingleses pronunciados por un mismo locutor de forma aislada.

Los primeros trabajos que emplearon medios informáticos aplicados al reconocimiento automático del habla aparecen en la década de los 60. Estos trabajos se centran en el problema del reconocimiento de palabras aisladas (pronunciadas con suficiente separación temporal) monolocutor (pronunciadas por un único locutor), y utilizaron técnicas de programación dinámica [Bellman57] para comparar parámetros de palabras con duraciones diferentes a través de un alineamiento temporal no lineal de las mismas.

A partir de este instante, comienza un gran aluvión de trabajos, principalmente orientados al reconocimiento de palabras aisladas, con la impresión optimista de poder extrapolar los resultados de forma que fuese posible lograr, en un corto periodo de tiempo, sistemas capaces de reconocer cualquier frase pronunciada por cualquier locutor de forma natural.

Con este objetivo, se abordaron grandes proyectos de investigación en el campo del reconocimiento automático del habla. De éstos, el más ambicioso y conocido de la historia del reconocimiento automático del habla se inició en 1971, financiado por el Departamento de Defensa de los EE.UU. Es el proyecto ARPA-SUR (Advanced Research Projects Agency - Speech Understanding System). Aunque los ambiciosos objetivos de éste y otros proyectos nunca llegaron a alcanzarse, contribuyeron en gran medida al conocimiento de los mecanismos de producción del habla, y a la toma de conciencia sobre la verdadera magnitud del problema, y la necesidad de la realización de investigación de base para la resolución de los problemas presentados.

1.1.1. PROBLEMAS RELACIONADOS

Varios tipos de problemas hacen difícil y no resuelto en la actualidad el problema del reconocimiento automático del habla.

Continuidad

No existe separación (silencio) entre las diferentes palabras que componen una frase, cuando ésta se pronuncia de forma natural. Esta característica establece una diferencia esencial entre el reconocimiento del habla, y el reconocimiento de textos escritos, donde el espacio actua como separador entre palabras.

Además, los sonidos elementales (fonemas) que componen el habla también aparecen concatenados sin ningún tipo de separación, y son modificados por su contexto inmediato, formado por los fonemas anterior y siguiente. Este fenómeno es conocido como "coarticulación", y es debido al hecho de que la pronunciación de un fonema requiere que el aparato fonador humano adquiera la configuración adecuada, proceso que conlleva una cierta inercia, de forma que no es posible una transición instantánea entre las configuraciones correspondientes a dos fonemas adyacentes.

Por último, también existen modificaciones más débiles de los fonemas debido a su posición dentro de la palabra debidas, por ejemplo,a la entonación.

Variabilidad

La voz humana presenta una gran cantidad de variabilidad, debida principalmente a tres causas diferentes, que hacen que una misma palabra nunca sea observada con exactamente las mismas características acústicas.

Parte de esta variabilidad es intra-locutor, debida a variaciones no lineales en las duraciones de los sonidos elementales, provocadas por el ritmo de pronunciación. También existe variablidad en la frecuencia e intensidad debidas a situaciones tales como el estado físico del locutor (cansancio, catarro,etc.) o al modo de pronunciación (cantando, gritando, etc.).

Una segunda fuente de variabilidad es inter-locutor. Esta es debida principalmente a variaciones físicas en los aparatos fonadores de los diferentes locutores, así como al sexo y edad de los mismos, que introducen variaciones en la escala de frecuencias. También existe una fuente de variabilidad relacionada con el modo de pronunciación y acento de los locutores.

Por último, el canal de transmisión y el entorno introducen una fuente adicional de variablidad (tipo de micrófono, ancho de banda de la línea de transmisión, interferencias, ruidos, etc.).

Debido a la variabilidad antes mencionada, es necesario observar una gran cantidad de datos relativos a los diferentes sonidos que componen la voz humana, a fin de extraer las características esenciales de éstos con independencia del contexto, entorno y locutor. En este sentido, un problema difícil para un sistema de reconocimiento es el de determinar si una /a/ pronunciada por un locutor adulto es más similar a una /a/ pronunciada por un niño o a una /o/ pronunciada por el mismo locutor.

Redundancia

Que la señal de voz es altamente redundante es sencillo de demostrar sin más que tener en cuenta que, mientras que el mensaje implícito en una comunicación oral normal puede transmitirse a una velocidad del orden de 50 bits/segundo, la transmisión adecuada de las señales de voz requiere del orden de 100 Kbits/segundo (p.e. 8 KHz y

12 bits/muestra).

Esta redundancia es debida a que la señal de voz transporta información adicional que identifica al locutor y su entorno, así como su estado de ánimo, modo de pronunciación, etc. Es por ésto que un sistema de reconocimiento debe focalizar su atención en la extracción de parámetros que caractericen adecuadamente el tipo de información útil para el proceso de reconocimiento.

Multiinteractividad

En el proceso de percepción y/o comprensión del habla, existen varios niveles fuertemente interrelacionados; cada uno de los cuales aporta información útil al proceso de reconocimiento. El *Nivel Acústivo*, correspondiente al análisis de las características físicas de la señal de voz. El *Nivel Fonético*, correspondiente a la determinación de los sonidos elementales (fonemas, sílabas, etc.). El *Nivel sintáctico*, en el que se extraen los elementos constitutivos (morfemas), y se aplican reglas gramaticales para la detección de construcciones correctas del lenguaje. El *Nivel semántico*, en el que se llega a la comprensión del mensaje transmitido, eliminando interpretaciones absurdas en base al conocimiento de la realidad y el contexto del mensaje recibido.

Todos estos niveles presentan fuertes interacciones entre si, y cada uno de ellos aporta información útil al proceso de reconimiento; sin embargo, no existe en la actualidad un formalismo que permita la integración e interpretación de las informaciones correspondientes a los diferentes niveles, haciendo compleja la tarea del reconocimiento del lenguaje natural.

1.1.2. RESTRICCIONES AL PROBLEMA

Debido a la complejidad de la tarea del reconocimiento del lenguaje natural hablado, ésta se ha abordado con diferentes grados de aproximación, determinados por diferentes condicionamientos o restricciones sobre la variablidad de la señal de voz a procesar, tanto respecto a l variabilidad inter-locutor como al ritmo y modo de pronunciación.

Monolocutor/multilocutor/independiente del locutor

Una primera restricción aplicada en los sistemas de reconocimiento se refiere a la limitación en variabilidad inter-locutor de la señal de voz. Así, un sistema de reconocimiento que reconoce voz de un único locutor de denomina *monolocutor*, mientras que los sistemas que admiten voz de diferentes locutores se suelen denominar *multilocutor*.

En la bibliografía, sin embargo, suele encontrarse una diferenciación entre sistemas *multilocutor*, y sistemas *independientes del locutor*. Usualmente, se reserva la denominación *multilocutor* para aquellos sistemas que admiten voz de entre un conjunto limitado de locutores, mientras que la denominación *independiente del locutor* se reserva para aquellos sistemas de reconocimiento en los que, a priori, se admite voz perteneciente a cualquier locutor.

La diferencia esencial entre los sistemas de reconocimiento multilocutor e independientes del locutor estriba en el conocimiento previo que el sistema posee de los locutores. En un sistema de reconocimiento multilocutor, es posible entrenar al sistema con las características de los locutores que componen el conjunto restringido antes citado, de forma que éste posee cierta información a priori sobre éstos. En un sistema independiente del locutor no es posible incorporar este tipo de información previa sobre los locutores, y normalmente, estos sistemas son evaluados utilizando un conjunto de locutores diferentes de los utilizados para el entrenamineto del mismo.

Palabras aisladas/conectadas y 'word-spotting'

El segundo tipo de restricción utilizada en el diseño de sistemas de reconocimiento de voz está relacionada con la forma en la que el/los locutores pronuncian las palabras.

La restricción más fuerte corresponde a los sistemas denominados de *palabras aisladas*. En tales sistemas, se condiciona al locutor a pronunciar las palabras con suficiente separación temporal (silencios) como para que sea posible la determinación de los límites entre éstas. En tales sistemas, la unidad de decisión suele ser la palabra, de forma que el proceso de reconocimiento se basa en la delimitación e identificación de las palabras pronunciadas por el locutor.

Un nivel inferior de condicionamiento corresponde a los sistemas de *palabras conectadas*, en los que el locutor puede pronunciar las palabras de forma fluida. Tales sistemas suelen utilizar la palabra como unidad de decisión, y basar su estrategia de reconocimiento en la segmentación e identificación conjuntas de las palabras que componen cada una de las frases pronunciadas por el locutor.

Una aproximación similar es la denominada "word-spotting", en la que el objetivo es la identificación de palabras correspondientes a un determinado vocabulario inmersas en frases en las que pueden aparecer otras palabras ajenas al mismo.

Por último, la denominación *voz contínua* se reserva para aquellos sistemas de reconocimiento que no imponen ningún tipo de restricción al modo de pronunciación de los locutores. Tales sistemas suelen utilizar unidades de decisión inferiores a la palabra (p.e. fonemas).

1.2. DIFERENTES APROXIMACIONES

Para abordar el problema del reconocimiento automático del habla se han utilizado diferentes aproximaciones. A continuación resumimos las principales.

1.2.1. COMPARACION DE PATRONES

La primera aproximación utilizada en reconocimiento automático del habla, e inicialmente aplicada a la tarea de reconocimiento de palabras aisladas monolocutor, es la técnica denominada *comparación de patrones* (*Pattern Matching* en la literatura inglesa).

Esta técnica consta de dos fases. En la primera de ellas, denominada fase de entrenamiento, el locutor pronuncia cada una de las palabras de un determinado vocabulario. Estas palabras son muestreadas y procesadas a nivel acústico extrayendo una secuencia temporal de vectores compuestos por coeficientes extraidos de un banco de filtros analógico o simulado a traves de FFT (Fast Fourier Transform. Transformada rápida de Fourier); o de un análisis autoregresivo LPC (Linear Prediction Coding. Codificación de Predicción Lineal); o de coeficientes derivados de los anteriores como el cepstrum de la señal. Estos coeficientes son extraidos a intervalos temporales regulares del orden de 10 milisegundos, con lo que la señal de voz queda caracterizada por una secuencia temporal de vectores que representan las características acústicas (espectrales) de la misma. Las

secuencias de vectores extraidas de esta forma se almacenan como prototipos o patrones de las palabras correspondientes.

En la fase de reconocimiento, las palabras pronunciadas por el locutor (palabras incógnita), son procesadas a nivel acústico en la forma descrita anteriormente, y las secuencias de vectores obtenidas son comparadas con los prototipos previamente almacenados, por medio de una medida de similitud espectral definida usualmente en base a una medida de "distancia" entre parejas de vectores. Distancias utilizadas para este propósito son por ejemplo la distancia euclídea en el caso de coeficientes extraidos de un banco de filtros o del cepstrum de la señal, o las distancias de Itakura o razón de semejanza en el caso de coeficientes LPC. El criterio de decisión utilizado se basa en determinar el patrón más similar a la palabra incógnita.

Dado que, como ya indicamos en la sección anterior, en general, las palabras incógnita no coinciden en longitud (número de vectores) con los patrones (ni siquiera con el correcto), es necesario realizar un alineamiento temporal de los vectores de la palabra incógnita con los de los patrones de forma que, cuando se compare ésta con el patrón adecuado, los vectores correspondientes al mismo sonido queden alineados, situación en la que se obtendrá la mínima distancia total (suma de las distancias entre los vectores alineados). Un algoritmo que permite realizar dicha alineación es el algoritmo de Programación Dinámica (DP en la literatura inglesa), más conocido como DTW (Dynamic Time Warping. Alineamiento Temporal Dinámico) en el contexto de reconocimiento de voz. Este algoritmo fue inicialmente desarrollado por Bellman [Bellman57], y aplicado por primera vez al reconocimiento de voz por Vintsjuk [Vintsjuk68] y Slutsker [Slutsker68]. El algoritmo permite obtener el alineamiento cuya suma de distancias entre parejas de vectores es mínima. El funcionamiento básico del algoritmo es como sigue.

Si definimos los valores $d(i,j)$ correspondientes a las distancias entre los vectores i-ésimo de la palabra incógnita, compuesta por I vectores y j-ésimo de la palabra patrón compuesta por J vectores, éstos forman una matriz como la esquematizada en la figura 1.1. El alineamiento de las dos secuencias de vectores se corresponde con un "camino" en la matriz de distancias (indicado con trazo grueso en la figura) que parte del elemento $d(1,1)$ y finaliza en el elemento $d(I,J)$. La distancia total entre los vectores alineados corresponde a la suma de los elementos de la matriz de distancias contenidos en el "camino", de forma que la búsqueda del alineamiento óptimo es equivalente a la búsqueda del camino con

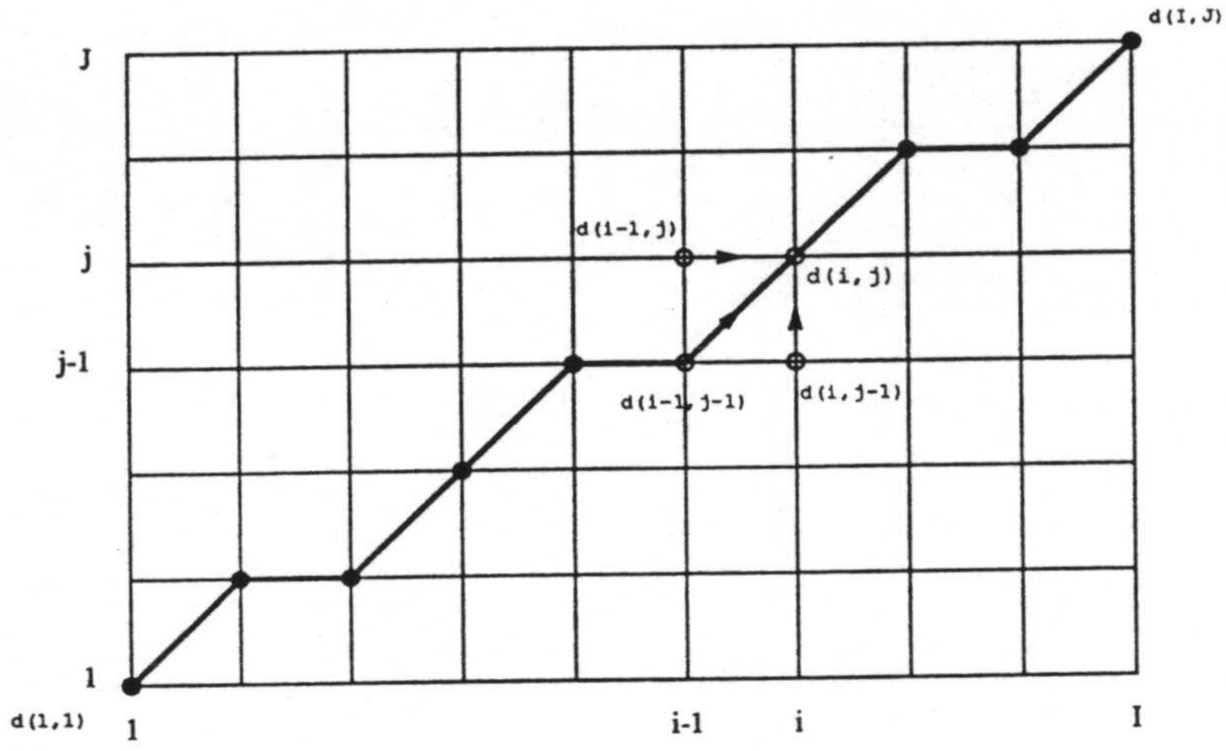

Figura 1.1. Algoritmo DTW

menor distancia total. Si definimos $g(i,j)$ como la distancia acumulada del camino óptimo que parte del elemento $d(1,1)$ y termina en el $d(i,j)$, se puede escribir la siguiente fórmula recursiva para la obtención de la distancia correspondiente a dicho camino.

$$g(1,1) = d(1,1) \tag{1.1a}$$

$$g(i,j) = \min \begin{cases} g(i-1,j) & + d(i,j) \\ g(i-1,j-1) & + d(i,j) \\ g(i,j-1) & + d(i,j) \end{cases} \tag{1.1b}$$

$$D = \frac{g(I,J)}{(I+J)} \tag{1.1c}$$

El valor D corresponde a la distancia media para el alineamiento óptimo de las dos palabras (incógnita y patrón), y se utiliza como criterio de decisión, seleccionando el patrón para el que dicha distancia media es mínima.

Extensión a múltiples locutores

La técnica básica de comparación de patrones, originalmente diseñada para sistemas monolocutor, se extendió a sistemas multilocutor e independientes del locutor almacenando varios patrones de referencia para cada una de las palabras del vocabulario. Estos patrones se seleccionan mediante técnicas de agrupamiento (*clustering* en la literartura inglesa) tales como el algoritmo K-medias [Wilpon85]. Opcionalmente, también se utilizan criterios de decisión más sofisticados como el K-NN (K Nearest Neighbours. K-vecinos más próximos) [Duda73].

Extensión a palabras conectadas

También se generalizó la técnica de comparación de patrones a sistemas de reconocimiento de palabras conectadas con algoritmos como el *Two Level DP Matching* [Shakoe79], el *Level Building* [Myers81] o el *One Pass DP* [Bridle82]. Básicamente, estos algoritmos determinan la secuencias óptima (o subóptima) de palabras (junto con sus límites) que corresponde a una cierta secuencia de palabras incógnita en general de longitud (número de palabras) desconocida, basándose en un criterio de mínima distancia.

Extensión a grandes vocabularios

En la extensión a grandes vocabularios de este tipo de sistemas, es necesario tener en cuenta sus grandes requerimientos de memoria y potencia de cálculo, dado que es necesario almacenar uno (o más) patrones de referencia para cada palabra del vocabulario, y realizar un proceso de comparación de cada palabra incógnita con cada uno de los patrones de referencia. A continuación expondremos algunas de las técnicas utilizadas para evitar estos problemas.

Cuantización vectorial

La técnica denominada cuantización vectorial o VQ (de la denominación inglesa Vector Quantization), fue originalmente desarrollada para codificación de voz a baja tasa de bits (*bit-rate*) [Linde80]. El método consiste en construir un conjunto de prototipos (diccionario) que represente adecuadamente el conjunto de posibles valores de los vectores

obtenidos para la representación de la señal de voz, y sustituir éstos vectores por los prototipos obtenidos, con un criterio de mínima distancia.

El método usual para la construcción del diccionario es la selección de un conjunto de prototipos (centros del diccionario) de forma que se minimice la distancia entre éstos y un amplio conjunto de vectores que componen el conjunto de entrenamiento.

Una vez construido el diccionario, los vectores correspondientes a los patrones de referencia son sustituidos por los centros del diccionario más similares (con menor distancia), con lo que se reduce el número de bits necesarios para almacenar cada vector de los patrones, ya que basta con almacenar el índice del centro correspondiente del diccionario. Asi, por ejemplo, con vectores de 10 componentes reales y un diccionario de 256 centros, el factor de reducción en el número de bits necesarios para almacenar los patrones es del orden de $\frac{10 \cdot 32}{8} = 40$.

En cuanto al coste computacional, si los vectores correspondientes a las palabras incógnita también son cuantizados en la forma descrita para los patrones de referencia, todas las distancias necesarias para el desarrollo del algoritmo DTW corresponden a distancias entre centros del diccionario, que se pueden almacenar precalculadas, con lo que no será necesaria la realización de ningún cálculo de distancia adicional, salvo los necesarios para la cuantización de los vectores de la palabra incógnita. Por ejemplo, si tenemos un vocabulario de N palabras de longitud media L y un diccionario con M centros, el número de distancias que es necesario calcular para la clasificación de una palabra incógnita de longitud L es símplemente $M \cdot L$, correspondientes a su cuantización. En el caso de no utilizar cuantización vectorial, el número de distancias necesarias para la clasificación sería $N \cdot L^2$, con lo que la reducción en el número de distancias a calcular es $\frac{N \cdot L^2}{M \cdot L}$. Por ejemplo, para un vocabulario de 128 palabras de longitud media 100 vectores y un diccionario de 256 centros, el factor de reducción sería 50.

Reconocimiento sin alineamiento temporal

Esta aproximación se basa en la hipótesis de que cada palabra posee un conjunto propio de sonidos, que la diferencian de las demás. Incluso cuando dos palabras contienen fonemas iguales o similares, el efecto de coarticulación hace que las características

acústicas de éstos sean diferentes, de forma que es posible, en principio, discriminar entre palabras de un vocabulario mediante un proceso de comparación de patrones sin tener en cuenta alineamiento temporal alguno.

A partir de esta idea y del concepto de cuantización vectorial, Burton y Shore [Shore83] desarrollaron un sistema de reconocimiento cuyo criterio de decisión se basa unicamente en las distorsiones de cuantización de los vectores de las palabras incógnita con diccionarios VQ correspondientes a las palabras del vocabulario.

Durante el entrenamiento, se construye un diccionario VQ para cada palabra del diccionario, que contiene prototipos de los vectores de características acústicas de las palabras. En la fase de reconocimiento, la secuencia de vectores correspondiente a una palabra incógnita, es cuantizada con los diccionarios VQ de las palabras del vocabulario, seleccionando el de menor distorsión media.

Sobre esta base común, se han realizado varias modificaciones para incorporar cierto tipo de información sobre el secuenciamiento temporal de los prototipos acústicos (centros del diccionario VQ) de las palabras. Algunas de estas variantes pueden encontrarse descritas en el capítulo 6, junto con una nueva aplicación que integra la información contenida en las distorsiones de cuantización con la probabilidad de generación de modelos HMM discretos en un contexto probabilístico.

Reconocimiento en dos pasos

Para acelerar el proceso de reconocimiento, se puede utilizar un esquema en el que un método menos preciso pero más rápido se utiliza para eliminar los candidatos más improbables reduciendo el número de comparaciones y disminuyendo el coste computacional del sistema.

Esta eliminación de candidatos puede realizarse calculando las distancias medias con los prototipos utilizando alineamiento lineal (sobre la diagonal de la matriz de distancias) [Gauvani86], o incluso sin tener en cuenta ningún tipo de alineamiento temporal [Mariani87].

Una vez eliminados los candidatos poco probables, se utiliza un algoritmo DTW o de otro tipo para la clasificación final sobre el conjunto de candidatos restante.

Unidades inferiores a la palabra

Otro método utilizado para reducir los requerimientos de memoria de los sistemas de reconocimiento es la utilización de unidades de decisión inferiores a la palabra, tales como fonemas [Sugamura83], difonemas [Schwartz80], sílabas [Watanabe83], demisílabas [Ruske81]; o unidades sin correspondencia lingüística, obtenidas mediante algoritmos de segmentación [Burton85a, Lee88a, Roucos82] y [Kopec85a, Bush87a, Lee86]; a las que se pueden aplicar técnicas similares a las utilizadas en reconocimento de palabras conectadas.

La reducción del número de prototipos a utilizar viene dada por el hecho de que mientras que un vocabulario puede estar formado por varios miles de palabras, sus unidades elementales (p.e. fonemas) no superan algunas decenas. El problema de la utilización de este tipo de unidades de decisión estriba en la variabilidad que presentan debido al contexto, como se indicó en la sección precedente.

1.2.2. MODELOS OCULTOS DE MARKOV

A diferencia del método de comparación de patrones, en el que una o más repeticiones de cada una de las unidades de decisión (palabras o unidades inferiores) son utilizadas como patrones de referencia en el proceso de reconocimiento, un modelo oculto de Markov (HMM, de la denominación inglesa Hidden Markov Model) representa un nivel superior de abstracción, en el que cada unidad de decisión es representada por un modelo. Esta aproximación fue utilizada por primera vez en reconocimiento de voz en CMU [Baker75] y en IBM [Jelinek76]. El reconocimiento se basa en la determinación del modelo más probable dada una secuencia de observaciones correspondiente a la unidad incógnita. Por ejemplo, si W es el modelo correspondiente a una unidad de decisión y A es la secuencia de observaciones correspondiente a una unidad incógnita, el proceso de reconocimiento se basa en la determinación del modelo para la que la probabilidad condicionada $P(W|A)$ es máxima. Esta probabilidad puede expresarse según la regla de Bayes en la forma siguiente

$$P(W|A) = \frac{P(A|W) \cdot P(W)}{P(A)}$$

donde $P(A)$ es la probabilidad incondicional de la secuencia de observaciones,

independiente del modelo considerado; $P(A|W)$ es la probabilidad a posteriori de la secuencia de observaciones A dado el modelo W; y $P(W)$ es la probabilidad incondicional del modelo. Si suponemos que todas las unidades de decisión son equiprobables, basta con evaluar $P(A|W)$ para cada modelo y seleccionar el de máxima probabilidad.

En el caso de voz continua, W representa una cadena de palabras (frase), y entonces $P(A|W)$ representa el modelo acústico y $P(W)$ el modelo de lenguaje. Los dos modelos pueden representarse con HMM's.

Aproximación discreta

En la aproximación discreta, cada unidad de decisión está representada por un autómata de estados finitos, compuesto por un conjunto de estados y un conjunto de arcos que los unen, representando las posibles transiciones entre estos. Asociados a los estados, existen unas funciones de probabilidad de producción (observación) de símbolos $b_i(k)$, correspondientes a un alfabeto finito, y que representan diferentes características acústicas de la señal de voz. Asociados a los arcos existen unas probabilidades a_{ij}, que determinan las transiciones entre parejas de estados del modelo.

En la figura 1.2 se muestra un autómata correspondiente a un modelo oculto de Markov. En un modelo de Markov de 1^{er} orden, y bajo la suposición de independencia de observaciones, tanto las probabilidades de transición a_{ij}, como las de producción de símbolos $b_i(k)$, dependen únicamente del estado actual S_i.

En algunas formulaciones las probabilidades de producción de símbolos se asocian a las transiciones en lugar de a los estados.

Dado que los vectores correspondientes a las características acústicas de la señal de voz son continuos, es necesario un proceso de "etiquetado" que asigne símbolos de entre un conjunto finito, con un cierto significado (acústico o fonético). Este proceso de etiquetado puede estar basado en técnicas de cuantización vectorial o similares, como el etiquetado microfonético [Andreu90] o los mapas fonotóticos [Kohonen84] basados en redes neuronales (ver siguiente sección).

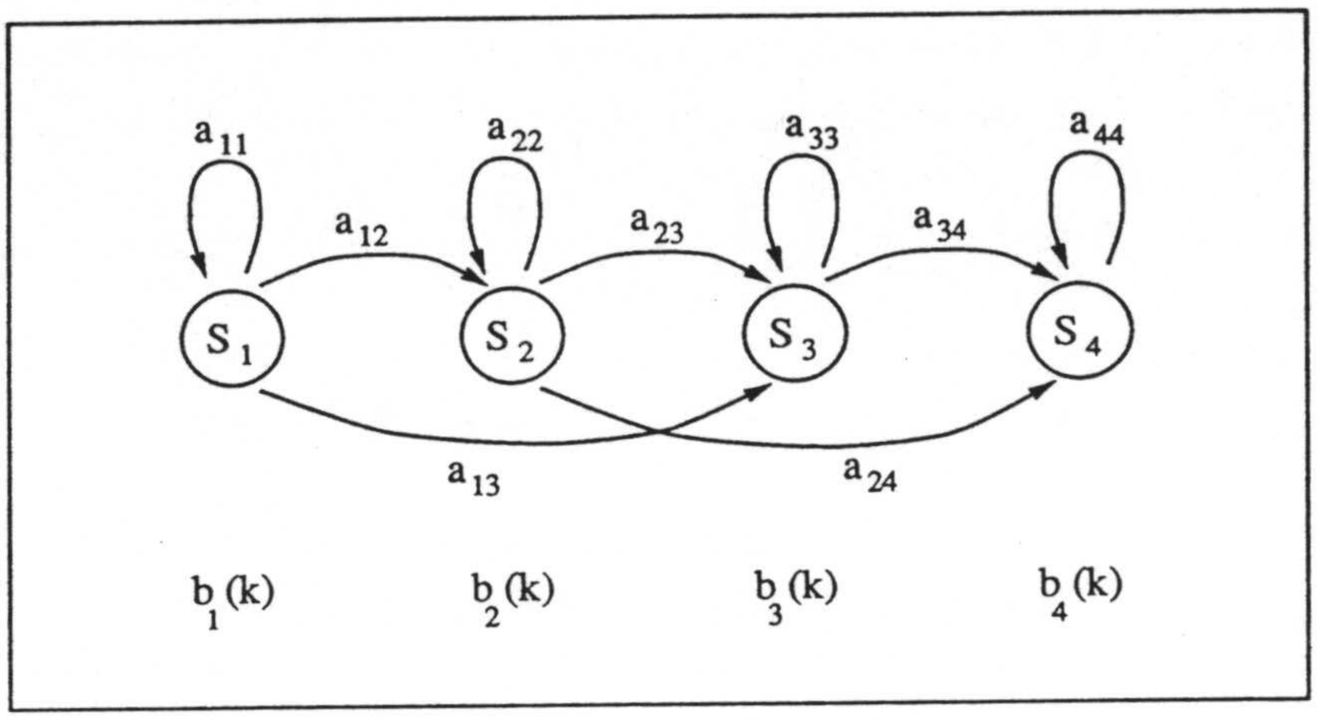

Figura 1.2. Modelo Oculto de Markov

Aproximación continua

En esta aproximación se mantienen las características de la anterior salvo en lo que se refiere a las probabilidades de producción de observaciones. En los modelos continuos, las observaciones del proceso son directamente los vectores continuos resultado del análisis acústico de la señal y, en consecuencia, las probabilidades de producción deben ser modeladas a través de funciones de densidad de probabilidad, en general multivariadas. Estas funciones de densidad de probabilidad son modeladas usualmente mediante gausianas multivariadas, o combinaciones lineales de éstas [Rabiner85a, Juang85].

Muchos de los aspectos relacionados con la utilización de los modelos de Markov para reconocimiento de voz pueden encontrarse en los capítulos 3 y 4 de la presente memoria, así como implementaciones de sistemas de reconocimiento basados en modelos discretos, en los capítulos 5 y 6.

1.2.3. APROXIMACION CONEXIONISTA

En esta aproximación, los datos de referencia son representados a través de patrones de actividad sobre una red de elementos de proceso sencillos interconectados.

Por la similitud con el funcionamiento del cerebro, a estas redes se les suele denominar *redes neuronales artificiales* (o simplemente *redes neuronales*), y a los elementos de proceso *neuronas*.

Perceptrones

El origen del perceptrón se encuentra en los trabajos de Rosenblatt [Rosenblatt59] sobre modelos de percepción visual, y fue abandonado porque se demostró [Minsky69] que no era capaz de sintetizar funciones sencillas como la XOR. Recientemente se ha recuperado este tipo de modelado debido, principalmente, a la utilización de perceptrones multicapa (MLP. Multilayer Perceptron), que no presentan la limitación antes mencionada [Lippman87], y al desarrollo de algoritmos de aprendizaje automático, como el *backpropagation* (retropropagación) [Rumelhart86]

Un perceptrón multicapa está compuesto por una serie de elementos de proceso, las neuronas, que implementan una función de activación sobre la suma pesada de sus entradas, y que están distribuídos en varios grupos o capas interconectadas entre sí, como se muestra en la figura 1.3. Dos capas interactúan con el exterior, la capa de entrada, que recibe los estímulos externos, y la capa de salida, que proporciona la respuesta de la red. Además existe una o más capas ocultas, inaccesibles desde el exterior.

Cada una de las capas está compuesta por varias neuronas, cuyas entradas están conectadas a las salidas de las neuronas de la capa inferior, a través de enlaces con pesos w_{ij}, positivos o negativos dependiendo de que el enlace correspondiente sea de excitación o inhibición.

El estímulo se introduce en la capa de entrada, y se propaga a través de las capas ocultas, hasta alcanzar la capa de salida, produciendo un determinado patrón de excitación en las neuronas de ésta.

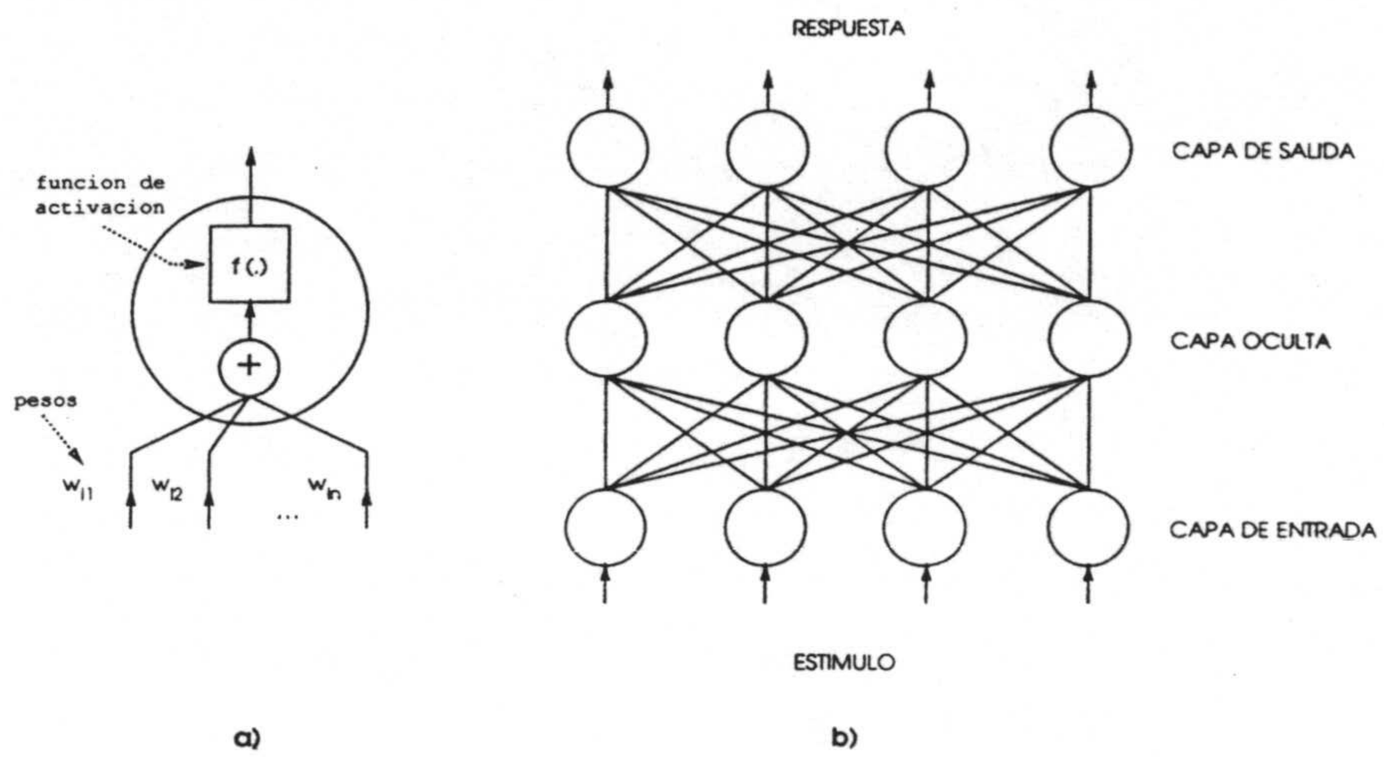

Figura 1.3. a) Neurona. b) Perceptrón multicapa

En la fase de entrenamiento, el patrón de excitación de la capa de salida es comparado con la respuesta deseada, generando una señal error, que es propagada hacia atrás en la red, reajustando los pesos w_{ij} de las diferentes conexiones entre capas. Este proceso se repite para un amplio conjunto de parejas estímulo-respuesta hasta conseguir el comportamiento deseado de la red.

Lo que se espera de las neuronas de las capas ocultas de la red, es que reorganicen sus conexiones de forma que la neurona correcta de la capa de salida sea excitada con un peso positivo alto y las incorrectas sean inhibidas con un peso negativo alto.

En la fase de reconocimiento, la neurona de la capa de salida con mayor nivel de activación designa el patrón a reconocer. En otras aproximaciones, el patrón de activación de la capa de salida se compara, mediante una medida de distancia (p.e. la distancia de Hamming, para neuronas con salida binaria), con los patrones de referencia.

Clasificación de patrones estáticos

Otras arquitecturas de redes neuronales se han utilizado para implementar procesos de agrupamiento no supervisado similares a la cuantización vectorial antes descrita.

Mapas de características

Los mapas de características o mapas fonotónicos [Kohonen84] se basan en la hipótesis de que, para el reconocimiento de voz, la información interrelacionada debe situarse topológicamente próxima, tal como ocurre en el cerebro humano [Kohonen84b].

Un mapa fonotónico tiene una arquitectura bidimiensional plana como la mostrada en la figura 1.4, en la que cada nodo corresponde a un prototipo espectral. Cada vez que un nuevo prototipo espectral se introduce en la red se localiza el prototipo más cercano, mediante una medida de distancia (p.e. euclídea) y se promedia. Los restantes nodos de la red son promediados con un peso que depende inversamente de la proximidad al nodo dado. De esta forma se consigue una configuración en la cual los prototipos similares se encuentran topológicamente próximos.

La máquina de Boltzman

La máquina de Boltzman es una red neuronal con todos los nodos organizados en una única capa y completamente conectados. Estos nodos están divididos en visibles e invisibles, y a su vez, los nodos visibles, en nodos de entrada y nodos de salida.

Cada nodo tiene asociada una probabilidad de encontrarse en el estado 0 ó 1, que depende de la diferencia de "energía" (suma pesada de las entradas procedentes de los demás nodos), que contiene un término asimilable al parámetro termodinámico temperatura

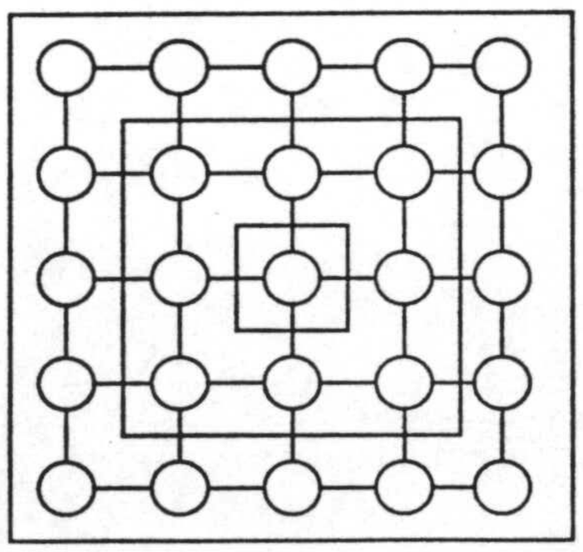

Figura 1.4. Arquitectura de un mapa fonotónico

(de ahí el nombre de la red). Cuanto mayor es la "temperatura" de la red, menor es la influencia mutua entre los distintos nodos. A medida que disminuye la temperatura, cada nodo está más afectado por el estado de los restantes nodos de la red.

En el proceso de entrenamiento, se asigna un valor aleatorio, 0 ó 1, a cada uno de los nodos, y una temperatura alta a la red. A continuación, se introducen los estímulos en los nodos de entrada, y se utiliza el patrón de activación de los nodos de salida, para modificar los pesos de las interconexiones. Una vez alcanzada una configuración estable para la red, el proceso se repite para un valor inferior de la temperatura de la red. La iteración de este proceso para valores decrecientes de la temperatura de la red origina que la red alcance un estado final de mínima energía, evitando la convergencia hacia mínimos locales. Este proceso suele denominarse *simulated annealing* (recocimiento simulado).

Existen técnicas similares desarrolladas para métodos de agrupamiento tradicionales, como el k-medias [Rose90a, Lu91], que se basan en el mismo concepto de recocimiento simulado.

Procesamiento temporal

Aunque las redes neuronales artificiales han mostrado buenas capacidades discriminativas para patrones estáticos, presentan problemas para modelar adecuadamente la evolución temporal inherente a las señales de voz.

Para solucionar este problema se han propuesto varias alternativas. La más sencilla es diseñar una capa de entrada con un número suficiente de neuronas como para acomodar la secuencia temporal de mayor longitud [Peeling88]; o bien, utilizar alguna técnica de compresión (lineal o no) de las señales temporales, para acomodar la duración de las secuencias temporales al tamaño de la capa de entrada de la red.

Redes neuronales con retardo temporal

Una posible aproximación al problema de la modelización temporal consiste en la utilización de retardos temporales, como en el TDMLP (Time Delay MultiLayer Perceptron. Perceptrón multicapa con retardo temporal) [Waibel89]. En la figura 1.5 se muestra el esquema de un TDMLP diseñado para el reconocimiento de consonantes. En el ejemplo considerado, la capa de entrada contiene 16 nodos alimentados por 16 coeficientes

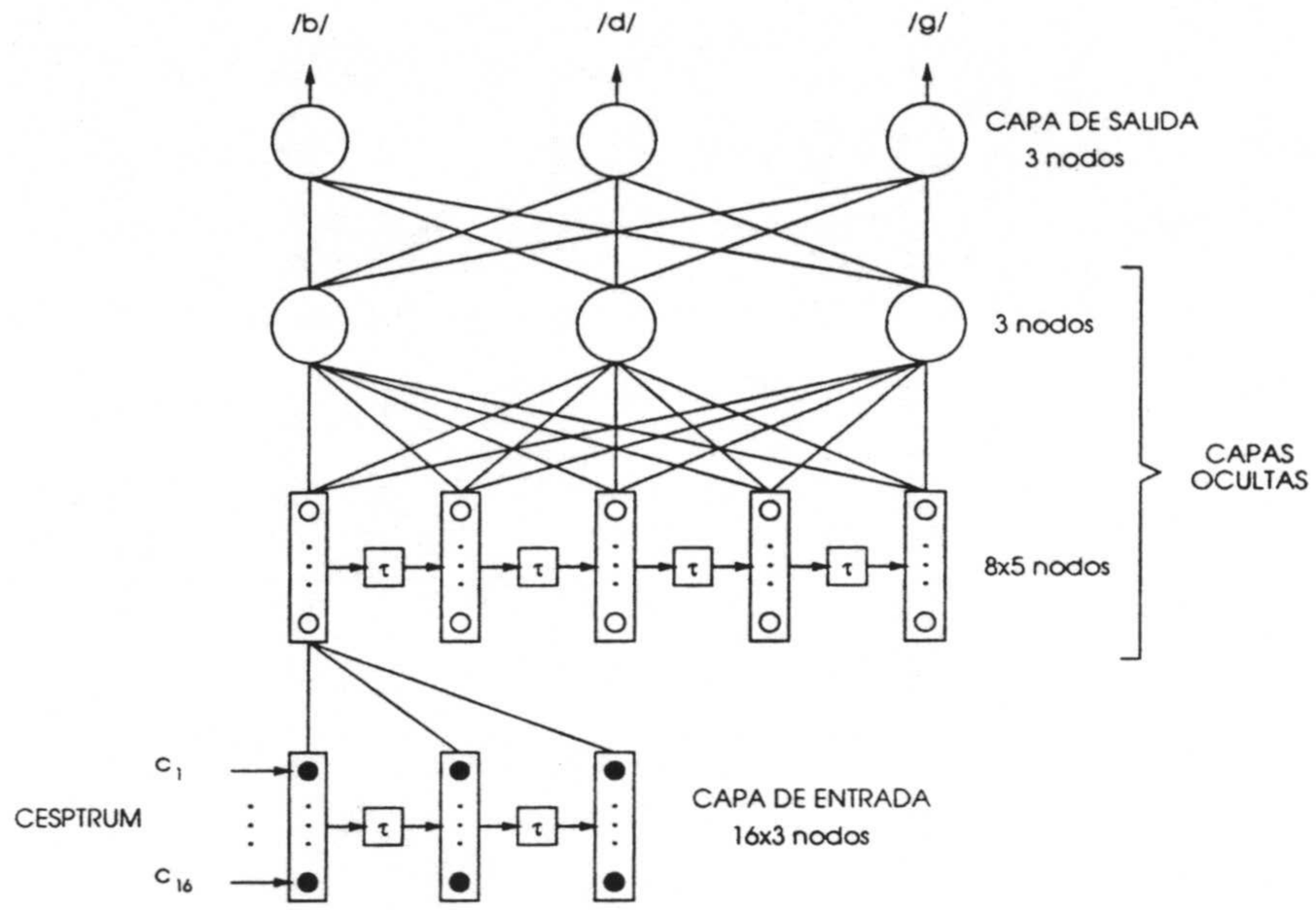

Figura 1.5. Esquema de un TDMLP

cepstrum, cuyas salidas se almacenan retardadas 10 y 20 ms, generando un contexto de entrada de 30 ms. Las salidas de la capa de entrada (3x16) alimentan a 8 nodos en la primera capa oculta, y las salidas de éstos se almacenan con retardos de 10, 20, 30 y 40 ms. Las salidas de la primera capa oculta (8x5) alimentan a 3 nodos en la segunda capa oculta, que, a su vez, excitan 3 nodos en la capa de salida, uno por cada consonante a reconocer (/b/, /d/ y /g/).

Esta técnica ha dado buenos resultados para la discriminación entre consonantes, pero, sin embargo, el coste computacional del entrenamiento de la red es muy elevado (varios días en un computador Alliant con 4 procesadores).

Redes Neuronales Recurrentes

Una aproximación alternativa al problema del procesamiento temporal es la propuesta para las redes neuronales recurrentes (RNR) [Bridle90a, Bourlard89].

Una red neuronal recurrente consta de una capa de entrada y otra de salida con conexiones recurrentes, según se muestra en la figura 1.6. Las neuronas de la capa de

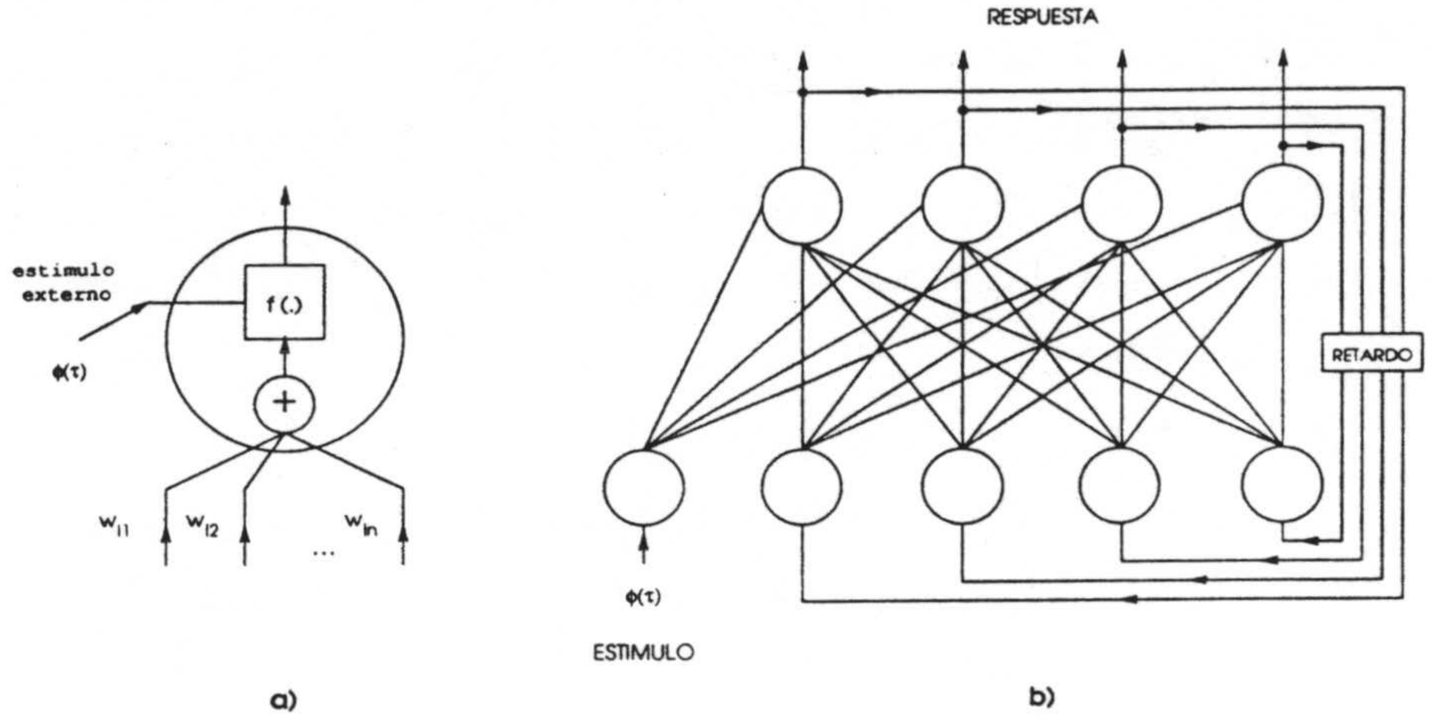

Figura 1.6. a) Neurona con estímulo externo. b) Red neuronal recurrente.

salida, además de la excitación proveniente de la capa de entrada, disponen de una entrada para un estímulo externo.

Las redes neuronales recurrentes se activan al recibir los estímulos externos $\phi(\tau)$ y los propagan cíclicamente por la red, a través de las conexiones recurrentes entre sus dos capas, de forma que la salida correspondiente a una secuencia de duración T aparece en la capa de salida tras haber procesado los T estímulos, siendo su comportamiento equivalente al de una red neuronal con T capas ocultas.

Las RNR presentan grandes analogías en su funcionamiento a los HMM [Bridle90a, Díaz91]

1.3. PRESENTACION Y JUSTIFICACION DEL TRABAJO

El trabajo abordado en la presente memoria se centra en el diseño de un sistema de reconocimiento de palabras aisladas independiente del locutor, para vocabulario reducido, con la orientación básica de obtener un sistema con una alta tasa de reconocimiento y requerimientos computacionales bajos.

Aunque en la actualidad muchos grupos de investigación comienzan a centrar sus trabajos en el reconocimiento de voz continua, sigue investigándose en sistemas de reconocimiento de palabras aisladas, debido a varias razones.

En primer lugar, el volumen de datos necesario para el entrenamiento y test de sistemas de reconocimiento de palabras aisladas es mucho más reducido que el requerido para el desarrollo de sistemas de reconocimiento de voz continua, y la potencia de cálculo necesaria para la realización de experimentos de reconocimiento en palabras aisladas es mucho menor que en el caso de voz continua, lo que hace que el diseño y test de tales sistemas sea mucho más asequible.

En segundo lugar, muchas de las técnicas utilizadas en el reconocimiento de voz continua (análisis acústico, modelado de unidades de decisión, algoritmos de entrenamiento y evaluación, etc.) han sido originalmente desarrollados sobre sistemas de reconocimiento de palabras aisladas y, posteriormente, trasladados con mínimas o nulas modificaciones a voz continua.

En tercer lugar, el problema del reconocimiento de palabras aisladas ha sido ampliamente estudiado, existiendo un conocimiento previo, y gran cantidad de resultados, que se pueden utilizar como referencia, lo que simplifica el problema de la evaluación comparativa de los resultados.

1.3.1. LA APROXIMACION SELECCIONADA

La aproximación seleccionada para el diseño del sistema de reconocimiento antes propuesto es la del modelado oculto de Markov discreto. Esta elección se debe al hecho de que los modelos discretos de Markov presentan un bajo coste computacional en comparación con las aproximaciones de comparación de patrones; y los resultados obtenidos en gran cantidad de trabajos indican un rendimiento comparable o superior al de los sitemas basados tanto en DTW como en redes neuronales.

Sin embargo, una de las limitaciones del modelado HMM discreto es el proceso de cuantización vectorial, que introduce errores en el proceso de asignación de símbolos a los vectores de características acústicas de la señal, lo que, a su vez, provoca un decremento en el rendimiento del sistema. Como alternativa al modelado discreto se han propuesto, al menos, dos modificaciones.

La primera de éstas consiste en la utilización de funciones densidad de probabilidad continua para modelar el proceso de producción de observaciones en los modelos, dando lugar a una variante denominada modelado continuo de Markov. Sin embargo, esta variante tiene el inconveniente del alto número de parámetros que es necesario estimar, lo que hace que, cuando el volumen de datos utilizado en la fase de entrenamiento no es suficientemente elevado (mucho más que en el caso discreto), los resultados finales obtenidos no sean superiores, e incluso sean inferiores, a los correspondientes al modelado discreto.

Una aproximación mixta es la formulada para los modelos semicontinuos de Markov [Huang90a, Huang89]. Estos aprovechan las ventajas del modelado continuo, limitando el número total de funciones densidad de probabilidad y, consecuentemente, el número de parámetros a estimar. Desde el punto de vista de la formulación continua, los modelos semicontinuos utilizan un conjunto común de gausianas en las mezclas que modelan las densidades de probabilidad de producción de símbolos de los diferentes estados de los modelos. Desde el punto de vista discreto, los modelos semicontinuos modifican el proceso de cuantización vectorial, modelando los centros del diccionario con gausianas multivariadas y generando múltiples candidatos en el proceso de cuantización. Al eliminar la restricción de clases disjuntas asumida en cuantización vectorial, se elimina, consecuentemente, la mayor parte de los errores introducidos en dicho proceso. Aunque los modelos semicontinuos reducen el coste computacional con respecto a los continuos al limitar el número de gausianas considerada, siguen teniendo un coste computacional del orden de 2 a 3 veces superior a los modelos discretos.

En el presente trabajo desarrollaremos una nueva variante del modelado HMM discreto cuya característica principal es la de utilizar diccionarios de cuantización vectorial independientes para cada uno de los modelos de las palabras del vocabulario. Esta variante está inspirada en los trabajos de Burton y Shore [Shore83], que mostraron como las distorsiones de cuantización sobre múltiples diccionarios VQ pueden utilizarse como criterio de clasificación, con buenos resultados. El inconveniente de esta aproximación es que no tiene en cuenta ningún tipo de información sobre el alineamiento temporal, dado que los patrones de referencia están únicamente representados por el conjunto de prototipos que forman el diccionario VQ. Varios trabajos han intentado solucionar este problema, así Pan [Pan85] y Furui [Furui88] incorporaron un postprocesador DTW para resolver las

situaciones en las que el proceso de cuantización no es capaz de realizar la discriminación. Burton [Burton85b] utilizó diccionarios correspondientes a diferentes secciones temporales de las palabras para incorporar cierto tipo de información sobre el alineamiento temporal de los prototipos espectrales, y Bergh [Bergh85] utilizó la distribución temporal de símbolos observados tras el proceso VQ con el mismo propósito.

En el presente trabajo, desarrollaremos un método que permite integrar las distorsiones de cuantización y las probabilidades de observación de símbolos de un modelo de Markov discreto (que modela el secuenciamiento temporal de los prototipos espectrales correspondientes a los centros del diccionario VQ) en un contexto probabilístico.

Cabe esperar que el proceso de cuantización realizado con diccionarios específicos para cada modelo sea más preciso que en el caso de utilizar un diccionario común a todos los modelos, al limitar el número de posibles centros para la cuantización con el diccionario de un modelo concreto. De otro lado, ésta aproximación permite considerar varias posibles cuantizaciones, una para cada palabra incógnita, y permite la utilización de información relativa al proceso de cuantización (distorsiones), e incorporarla al proceso de decisión. De esta forma, la determinación de la cuantización óptima se realiza simultaneamente con el proceso de decisión y no a priori como en el caso de los modelos discretos estandar.

En el presente trabajo se han construído tres sistemas de reconocimiento basados en modelos ocultos de Markov. Un primer sistema basado en modelos HMM discretos estándar, un segundo sistema basado en modelos semicontinuos, y un tercero basado en la nueva variante del modelado discreto antes expuesta, y se han obtenido resultados comparativos que muestran que el sistema propuesto ofrece mejores tasas de reconocimiento que los anteriores para configuraciones con igual número de prototipos en el proceso de cuantización vectorial, con requerimientos computacionales idénticos a los correspondientes al modelado discreto estándar.

1.4. ESQUEMA DE LA MEMORIA

Los restantes capítulos de la presente memoria están organizados como se indica a continuación.

En el capítulo 2 se describe el diseño y las características de la base de datos utilizada en el presente trabajo.

En el capítulo 3 se hace una introducción general al modelado HMM y se revisan las soluciones a los tres problemas básicos: entrenamiento, evaluación y decodificación; tanto para el modelado discreto como continuo. También se discuten cuestiones relativas a la implementación del modelado HMM tales como el escalado de las probabilidades, para evitar rebosamientos en la representación numérica, la utilización de múltiples secuencias de observaciones en el entrenamiento; y el problema del entrenamiento insuficiente.

En el capítulo 4 se describen cuestiones relacionadas con la utilización de los modelos HMM en reconocimiento de voz. Se describe la representación paramétrica utilizada para la señal de voz y el proceso de cuantización vectorial, así como la estimación inicial de los parámetros de los modelos.

En el capítulo 5 se describe una primera implementación del sistema de reconocimiento con modelos HMM discretos estándar, así como la incorporación de características dinámicas del espectro de la señal a la representación paramétrica de la misma. También diferentes técnicas de ponderación espectral son incorporadas en la función distancia del cuantizador vectorial. Por último se discute la introducción de información sobre la duración de los estados de los modelos para minimizar los efectos de la suposición incorrecta de distribución de probabilidad de duración de estados exponenciales, implícita al modelado HMM.

En el capítulo 6 se presenta la implementación del sistema de reconocimiento basado en HMM semicontínuos, así como experimentos para la selección del número de candidatos considerado en el proceso de cuantización vectorial y los resultados obtenidos. En este capítulo también se presenta el desarrollo de los modelos discretos con diccionarios VQ específicos. Se realiza una descripción formal de esta nueva variante del modelado discreto, presentando las fórmulas de evaluación de probabilidades y discutiendo

las particularidades relativas a los procesos de entrenamiento y evaluación de este tipo de modelos. Se presentan los resultados obtenidos, en comparación con los correspondientes al modelado discreto y semicontinuo. Por último, se describe un método para la reducción del número de centros totales utilizados en los diccionarios VQ de los modelos, basado en un proceso de agrupamiento que fuerza a que parte de los centros de los diccionarios VQ sean compartidos por varios modelos.

CAPITULO 2

LA BASE DE DATOS

2.1. DESCRIPCION DE LA BASE DE DATOS

Para el desarrollo de la tarea de reconocimiento propuesta es necesario disponer de un amplio conjunto de muestras de las diferentes palabras del vocabulario pronunciadas por varios locutores, a fin de disponer de datos suficientes para el entrenamiento y test del sistema. Por este motivo, se ha diseñado y contruido una base de datos con múltiples locutores, cuyas características principales pasamos a describir a continuación.

El diseño de la base de datos se basa en otra previamente desarrollada por el grupo de investigación al que pertenece el autor [Peinado89], y que contiene diez repeticiones, pronunciadas por un único locutor, de un vocabulario compuesto por seis palabras clave correspondientes a posibles designaciones de los motores eléctricos que controlan las articulaciones de un robot didáctico (Teach-2000). Esta base de datos se construyó para el desarrollo de un sistema de control de dicho robot mediante órdenes orales.

La base de datos desarrollada en este trabajo constituye una extensión de la anteriormente descrita en dos aspectos diferentes. En primer lugar, se ha extendido el vocabulario incorporando los diez dígitos castellanos al mismo y, en segundo lugar, se ha incrementado el número de locutores a cuarenta, de forma que pueda ser utilizada en tareas de reconocimiento multilocutor y locutor-independiente.

2.2. VOCABULARIO

Como ya indicamos anteriormente, el vocabulario de la base de datos está compuesto por dos conjuntos de palabras, listados en la tabla siguiente.

DIGITOS	PALABRAS CLAVE
CERO	CUERPO
UNO	HOMBRO
DOS	CODO
TRES	MUÑECA
CUATRO	MANO
CINCO	DEDOS
SEIS	
SIETE	
OCHO	
NUEVE	

Vocabulario de la base de datos

El primer conjunto de palabras está formado por los dígitos castellanos (0-9), y el segundo por un conjunto de palabras clave correspondientes a las denominaciones de las articulaciones del robot.

El vocabulario seleccionado es similar en tamaño y composición a otros utilizados en tareas de reconocimiento de palabras aisladas tales como el de los dígitos ingleses utilizado en AT&T [Rabiner85], el de dígitos castellanos utilizado en la Universidad de Valencia [Casacuberta90] y otros [Huang90b, Fissore90].

2.3. LOCUTORES Y REPETICIONES

En la actualidad, el número de locutores que componen la base de datos es de 40, 20 masculinos y 20 femeninos. La selección de los locutores se realizó atendiendo a un criterio de disponibilidad, entre profesores y alumnos del Departamento al que pertenece el autor. De esta forma, todos ellos son profesores o alumnos universitarios.

La procedencia geográfica de los locutores es esencialmente de la Comunidad Andaluza, de forma que se incluyen locutores de la mayor parte de las provincias que la componen. Ocasionalmente, se incluyen locutores de otras Comunidades Autónomas.

Los locutores han pronunciado tres repeticiones de cada una de las palabras del vocabulario, sin entrenamiento previo, es decir, de forma expontánea.

La grabación se realizó de forma que cada locutor pronunció las 16 palabras del vocabulario en tres series de grabaciones consecutivas, de esta forma se obtuvieron 120 repeticiones de cada palabra, correspondientes a 3 repeticiones pronunciadas por cada uno de los 40 locutores.

2.4. ADQUISICION DE DATOS

La adquisición de datos se realizó utilizando un sistema desarrollado por nuestro grupo de investigación, basado en un microordenador IBM-PC con una tarjeta de conversión A/D D/A ADA-PC14, y un banco de filtros analógico para la limitación del ancho de banda de la señal de voz.

La tarjeta de conversión está controlada por el programa ADAMENU, que permite el muestreo y conversión de señales, así como su almacenamiento y recuperación de disco; y que incorpora comprobaciones sobre el rango de la señal muestreada para detectar señales saturadas o con amplitud insuficiente. El programa permite además la visualización y delimitación automática de las palabras muestreadas así como la edición de los límites obtenidos en el proceso de delimitación.

2.4.1. CONDICIONES DE GRABACION

La grabación de las palabras se realizó utilizando un micrófono de ambiente y un amplificador de alta fidelidad, en un entorno de laboratorio. La relación señal/ruido media de las palabras de la base de datos es aproximadamente 24 decibelios.

La frecuencia de muestreo utilizada fue de 8 KHz, con una limitación de banda en el rango 60 Hz a 3800 Hz. La limitación se realizó mediante una pareja de filtros analógicos Butterworth, uno de sexto orden para el filtro pasa alta de 60 Hz, y uno de octavo orden para el filtro pasa baja de 3800 Hz.

La representación digital de las señales se realizó utilizando 12 bits por muestra en formato de complemento a dos, obteniendo un rango de valores [-2047,+2048].

2.4.2. DELIMITACION DE LAS PALABRAS

Las palabras, una vez digitalizadas, se delimitaron para eliminar los silencios inicial y final con el que éstas son grabadas. Esta delimitación se realizó de forma automática con el programa ADAMENU, que implementa un algoritmo implícito de delimitación que es una variante [Segura84] del propuesto por Rabiner y Sambur [Rabiner75].

Los límites obtenidos por el algoritmo son visualizados y opcionalmente editados manualmente para corregir errores graves de delimitación, correspondientes a la exclusión de una parte significativa de la palabra a delimitar.

El algoritmo de delimitación

Una descripción detallada del algoritmo de delimitación utilizado en la construcción de la base de datos descrita se puede encontrar en [Segura84a, Peinado89], a continuación describiremos únicamente las generalidades del mismo.

El algoritmo procesa la señal digitalizada extrayendo una serie consecutiva de segmentos de la misma obtenidos desplazando una ventana de Hamming de 32 ms de duración, en incrementos de 16 ms, sobre la señal. Los segmentos así obtenidos son preenfatizados, para eliminar posibles niveles de contínua, con un filtro digital de función de transferencia $H(z) = 1 - 0.95\,z^{-1}$, y a partir de éstos se evaluan las funciones energía y pasos por cero, correspondientes a las secuencias de valores de la energía y el número de cruces por cero de los segmentos de señal.

Estas dos funciones temporales son utilizadas por el algoritmo de delimitación en la forma siguiente. En primer lugar, se utilizan los 192 primeros milisegundos (12 primeros segmentos) de la señal, que se asumen correspondientes al silencio inicial de la palabra, para estimar las características del silencio, para lo que se evaluan los valores medios de la energía y los cruces por cero, así como la desviación típica de este último valor. El algoritmo también extrae el valor máximo de la energía de los segmentos de señal sobre la totalidad de la señal.

Con estos parámetros, se calculan tres umbrales. Un umbral superior de energía *UESUP*, que se elige de forma que se asegure que todos los segmentos de señal con energía superior a este valor correspondan con certeza a la palabra y no a los silencios inicial o final. Un umbral inferior de energía *UEINF* y un umbral de pasos por cero *UZS*, determinados de forma que los segmentos de señal correspondientes al silencio presenten energías superiores a *UEINF* o un número de pasos por cero superior a *UZS* (para consonantes fricativas con baja energía pero con componentes de alta frecuencia).

Una vez determinados estos tres umbrales, el algoritmo procede de la siguiente forma para la delimitación del inicio de la palabra. En primer lugar, se procesan los segmentos de señal desde en primero hacia adelante, hasta encontrar uno cuya energía supera el umbral *UESUP*. A partir de este punto, el algoritmo retrocede hasta encontrar un segmento cuya energía no supera el umbral *UEINF*, ni su número de cruces por cero es superior al umbral *UZS*. En este punto el algoritmo decide la colocación del límite inicial de la palabra. La delimitación del final de la palabra se realiza de forma simétrica a partir de último segmento de la misma.

El algoritmo también incorpora lógica adicional para evitar delimitaciones incorrectas como las que provocan silencios que ocasionalmente pueden aparecer en determinadas palabras. Estos corresponden la mayoría de las veces a los silencios previos a las consonantes plosivas.

En la siguiente figura se muestra esquemáticamente el funcionamiento del algoritmo en la delimitación de la palabra /SEIS/. La línea contínua representa la energía logarítmica, y la discontínua el número de pasos por cero.

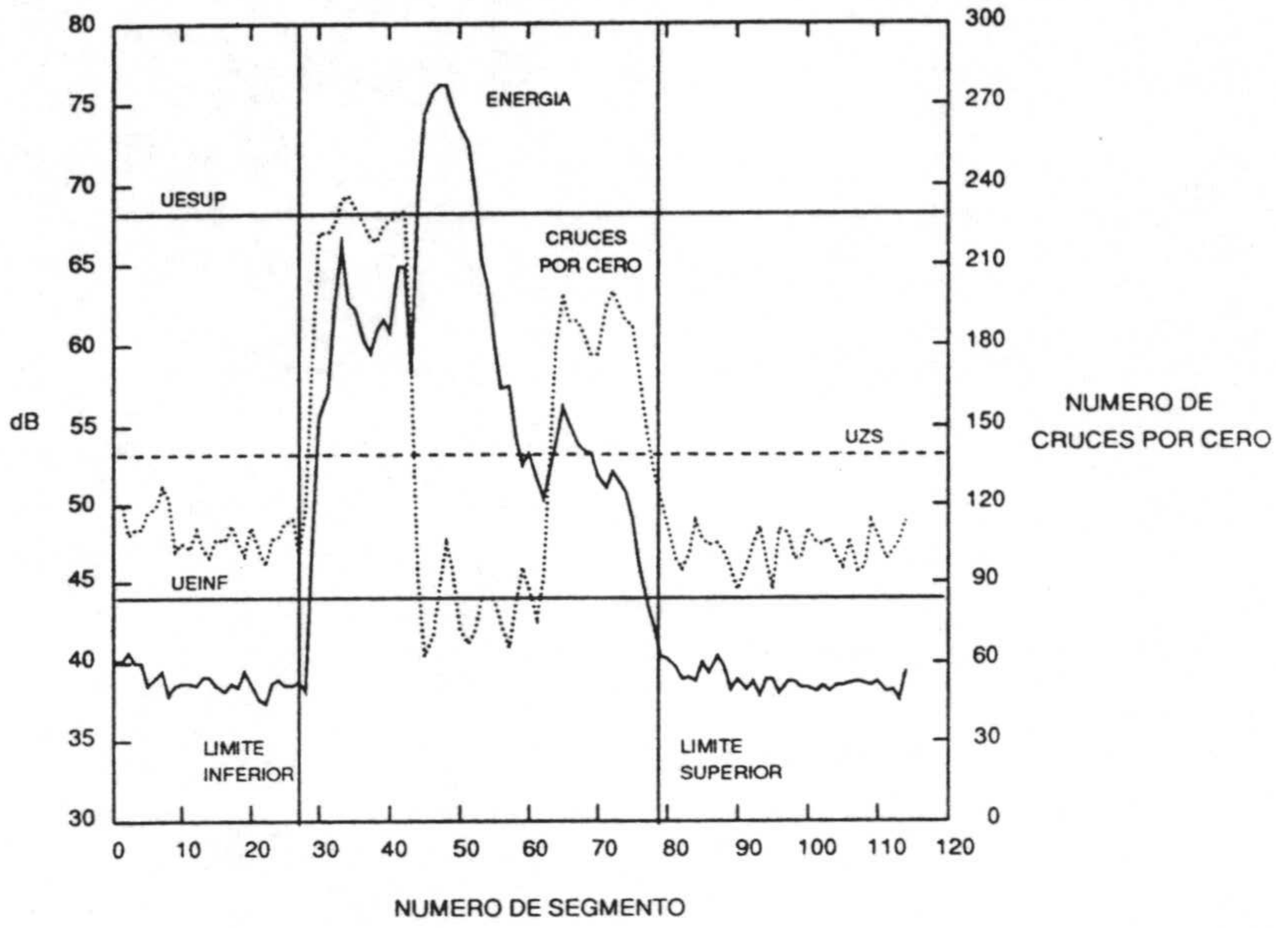

Figura 2.1. Delimitación de /SEIS/

2.5. RESUMEN DE CARACTERISTICAS

A continuación resumimos las características principales de las base de datos.

VOCABULARIO	16 palabras. Dígitos (0-9) y 6 palabras clave.
LOCUTORES	40 locutores. 20 masculinos y 20 femeninos.
REPETICIONES	1920 palabras. 120 repeticiones de cada palabra, 3 por locutor.
MUESTREO	Frecuencia 8 KHz. Ancho de banda [60 Hz, 3800 Hz]. Representación 12 bits en complemento a 2.
RELACION SEÑAL/RUIDO	23.95 dB.

CAPITULO 3

MODELOS OCULTOS DE MARKOV

3.1. PROCESOS DE MARKOV

En este apartado introduciremos el concepto de proceso de Markov como paso previo a la introducción de un concepto más general de modelado de procesos estocásticos, los Modelos Ocultos de Markov.

3.1.1. LA PROPIEDAD DE MARKOV

Un proceso de Markov se define como un proceso estocástico [Kampen81] en el que, para cualquier secuencia de instantes de tiempo de la forma $t_1<t_2<\cdots<t_n$, se verifica

$$P_{1|n-1}(y_n,t_n|y_1,t_1;y_2,t_2;\cdots;y_{n-1},t_{n-1}) = P_{1|1}(y_n,t_n|y_{n-1},t_{n-1}) \tag{3.1}$$

Es decir, que la densidad de probabilidad condicional de la variable estocástica Y en el instante de tiempo t_n dado el valor y_{n-1} de la variable en el instante de tiempo anterior t_{n-1} está definida de forma única y es independiente del conocimiento de los valores de la variable en instantes instantes de tiempo anteriores a t_{n-1}.

Un proceso de Markov está determinado de forma única por el conocimiento de las funciones densidad de probabilidad

$$P_1(y_1,t_1) \tag{3.2a}$$

$$P_{1|1}(y_2,t_2|y_1,t_1) \tag{3.2b}$$

que determinan la función densidad de probabilidad inicial y condicionada respectivamente.

Esta última se denomina *probabilidad de transición.* Conocidas éstas densidades de probabilidad es posible reconstruir la jerarquía completa del proceso, a través de la secuencia de instantes de tiempo, en la forma

$$P_{1|n-1}(y_n,t_n|y_1,t_1;y_2,t_2;\cdots;y_{n-1},t_{n-1}) = P_1(y_1,t_1)\prod_{m=2}^{n} P_{1|1}(y_m,t_m|y_{m-1},t_{m-1}) \quad (3.3)$$

Donde P_1 representa la densidad de probabilidad incondicional de un suceso y $P_{1|n}$ representa la densidad de probabilidad de un suceso condicionado a un conjunto de n sucesos. Esta propiedad es la que hace manejables a los procesos de Markov, y es la razón de su utilidad en múltiples aplicaciones.

Hay múltiples ejemplos de procesos físicos que verifican la propiedad de Markov, y por tanto pueden considerarse como procesos de Markov. El más antiguo conocido es el del movimiento Browniano, pero también lo son el decaimiento en la actividad radiactiva de sustancias o la disociación de las moléculas de un gas binario [Kampen81].

La propiedad de Markov puede aplicarse a un gran número de situaciones en las que la variable estocástica que determina el proceso puede ser contínua o discreta, univariada o multivariada, y la variable tiempo discreta o contínua.

3.1.2. CADENAS DE MARKOV

Una clase especial de procesos de Markov de particular interés es la constituida por procesos de Markov en los que, tanto la variable tiempo como la variable estocástica del proceso son discretas. Este tipo de procesos se denominan *Cadenas de Markov*. Una cadena de Markov es un proceso de Markov que verifica las siguientes propiedades

1) El rango de la variable estocástica Y es un conjunto discreto de estados.

2) La variable tiempo es discreta, y toma valores enteros.

$$t = \cdots,-2,-1,0,1,2,\cdots$$

3) El proceso es estacionario o al menos homogéneo en el tiempo, de forma que las probabilidades de transición dependen únicamente de la longitud del intervalo de tiempo considerado y no de los instantes de tiempo que lo delimitan.

Con respecto a la última restricción, en la bibliografía se pueden encontrar definiciones de cadenas de Markov en las que no figura la exigencia de homogeneidad temporal, denominando *Cadenas Homogéneas de Markov* a aquellas que verifican dicha propiedad.

Por último, denominaremos *Cadenas Finitas de Markov* a aquellas en las que el rango de la variable estocástica sea finito.

Un ejemplo de este tipo de procesos son las denominadas *Fuentes de Markov de primer orden* [Abramson74], que son modelos de fuentes de información en los que se supone que el símbolo emitido por la fuente depende del emitido en el instante de tiempo anterior. En este caso, el modelo del proceso de emisión de símbolos constituye una cadena de Markov para la que el último símbolo emitido constituye el estado de la fuente, de esta forma, el conjunto de estados $S=\{s_i\}_{i=1..n}$ coincide con el alfabeto de la fuente, y la propiedad de Markov establece que

$$P(y_t | Y_1^{t-1}) = P(y_t | y_{t-1}) \tag{3.4}$$

donde $Y_1^{t-1} = y_1 y_2 \cdots y_{t-1}$ define la secuencia de símbolos generados por la fuente. Las probabilidades de transición a_{ij} quedan definidas en la forma

$$a_{ij} = P(y_t = s_j | y_{t-1} = s_i) \qquad 1 \le i, j \le n \tag{3.5a}$$

$$\sum_{j=1}^{n} a_{ij} = 1 \qquad 1 \le i \le n \tag{3.5b}$$

y son, por la propiedad de homogeneidad temporal, independientes del instante de tiempo considerado.

En la figura 3.1 se muestra un modelo para una fuente de Markov de primer orden con alfabeto fuente ternario.

El concepto de fuente de información de Markov puede extenderse a situaciones en las que la dependencia en los símbolos emitidos se extiende a un número finito m de símbolos emitidos por la fuente con anterioridad, en cuyo caso se habla de *Fuentes de Markov de orden m*, para las que se puede escribir una relación similar a la que verifican las de primer orden en la forma

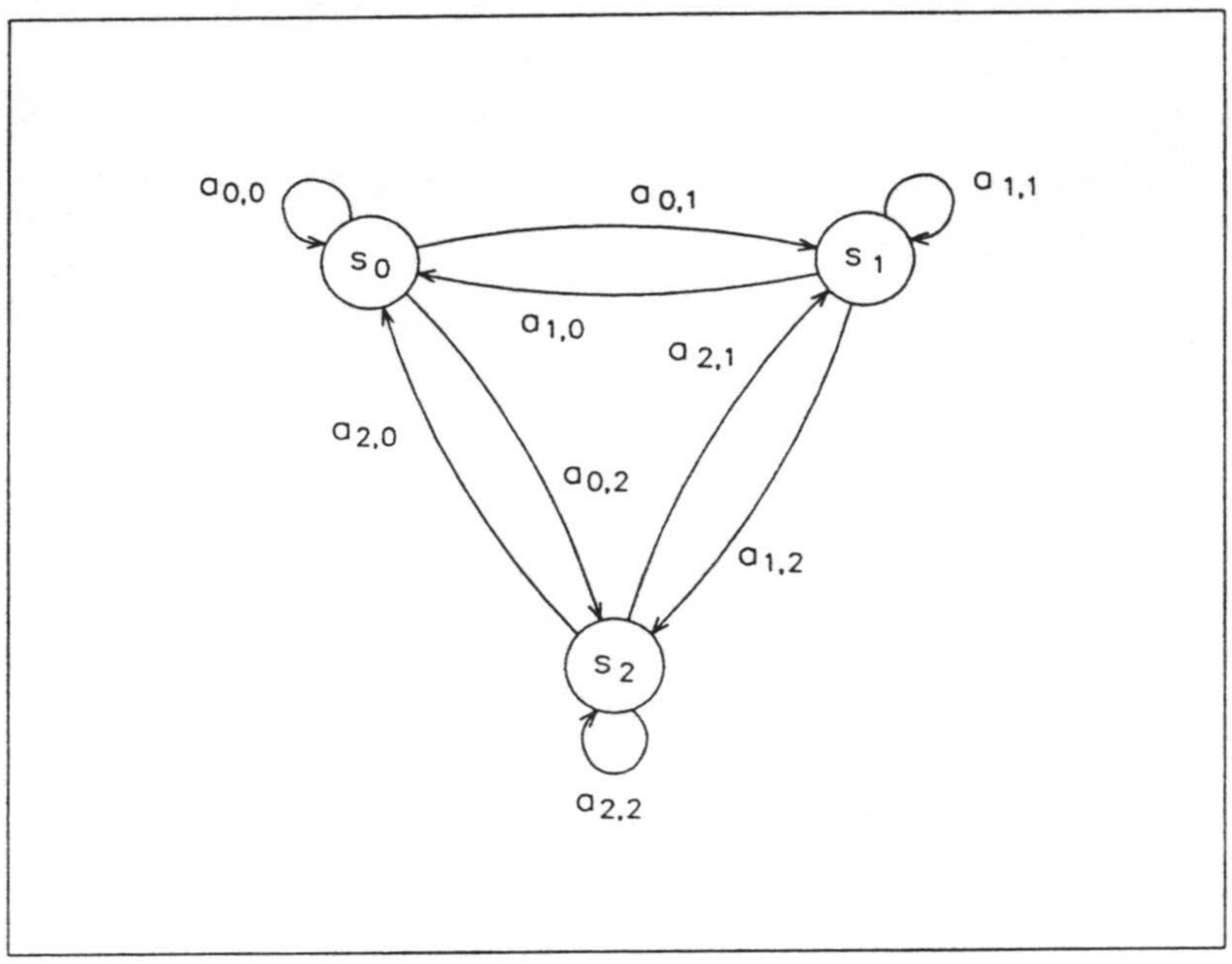

Figura 3.1. Fuente de Markov ternaria de primer orden

$$P(y_t \mid Y_1^{t-1}) = P(y_t \mid Y_{t-m}^{t-1}) \tag{3.6}$$

En este caso, los estados están constituidos por las n^m combinaciones de los m últimos símbolos emitidos por la fuente

$$S = \{\sigma_i\} \qquad 1 \leq i \leq n^m \tag{3.7a}$$

$$\sigma_i = s_{i_1} s_{i_2} \cdots s_{i_m} \tag{3.7b}$$

En este caso, no todas las transiciones entre estados están permitidas como se muestra en la figura 3.2, sino que sólo las $n^m \cdot n$ transiciones provocadas por la generación de cada uno de los n símbolos en los n^m estados pueden ser distintas de cero. Puede apreciarse de la comparación de los dos ejemplos antes descritos que no existe diferencia cualitativa entre los modelos de las dos fuentes de información, más que en la topología del modelo, controlada por los valores de las probabilidades de transición.

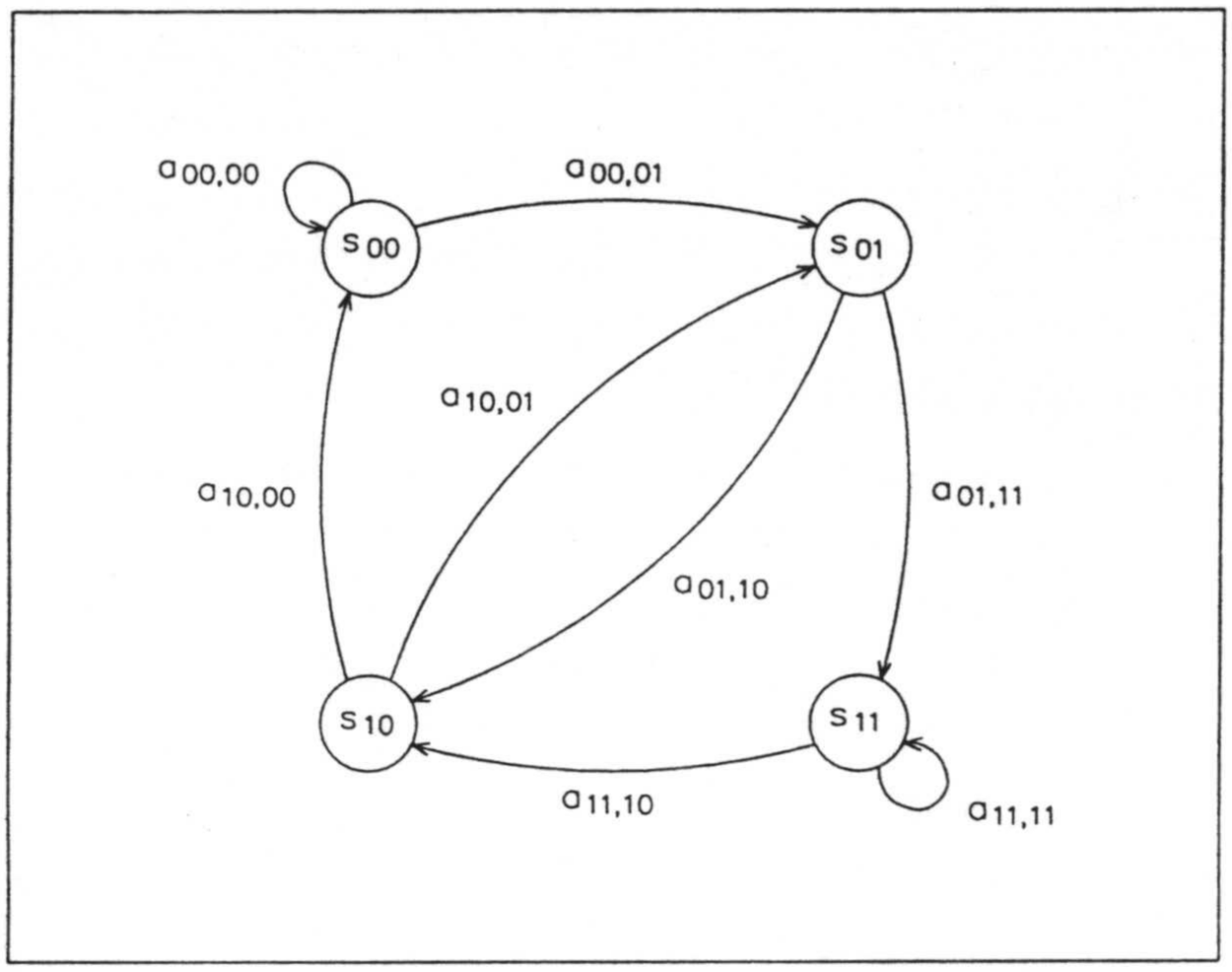

Figura 3.2. Fuente de información de Markov binaria de segundo orden.

3.2. MODELOS OCULTOS DE MARKOV

Los modelos de Markov antes introducidos han sido aplicados a una gran variedad de situaciones de forma satisfactoria; sin embargo conllevan una restricción implícita en cuanto a que el estado del proceso debe ser observable, condición demasiado restrictiva para determinados tipos de procesos. A continuación extenderemos el concepto de Modelo de Markov para englobar aquellas situaciones en las que el estado del sistema a describir es sólo observable de forma indirecta a través de una varibale probabilísticamente asociada al mismo. El modelo resultante lo denominaremos, en adelante, *Modelo Oculto de Markov*, o HMM (de las iniciales de la denominación inglesa *Hidden Markov Model*).

Esta denominación alude al hecho de que en un HMM, el proceso a modelar está compuesto por dos procesos estocásticos, uno correspondiente a la secuencia de estados a través de los cuales evoluciona temporalmente el sistema, que no es observable directamente (permanece oculto), y otro, asociado al anterior, que produce las observaciones y que depende unicamente del estado actual del sistema.

Para fijar ideas expondremos a continuación un ejemplo clásico [Rabiner89] que corresponde a un experimento de extracción de bolas coloreadas de un conjunto de urnas.

Supongamos que se dispone de un conjunto de N=4 urnas cada una de las cuales contiene bolas de M=3 colores diferentes en diferentes proporciones, y supongamos además que una persona extrae una serie de bolas de estas urnas en la forma siguiente: Primero se elige una urna con un determinado criterio, a continuación se extrae una bola de esta urna; una vez hecho ésto, se elige la urna de la que se extraerá la siguiente bola basándose en un criterio que depende únicamente de la urna actualmente seleccionada, y de ésta se extrae una nueva bola. El proceso de selección/extracción se repite hasta haber extraido una secuencia de T bolas.

Si denominamos $\{U_1,U_2,U_3,U_4\}$ al conjunto de urnas y $\{R\ ,V\ ,A\}$ al conjunto de tipos (colores) de bolas, la secuencia de colores obtenida será, por ejemplo, de la forma

$$X = \{R\ ,V\ ,R\ ,A\ ,A\ ,\ \cdots\ ,V\}$$

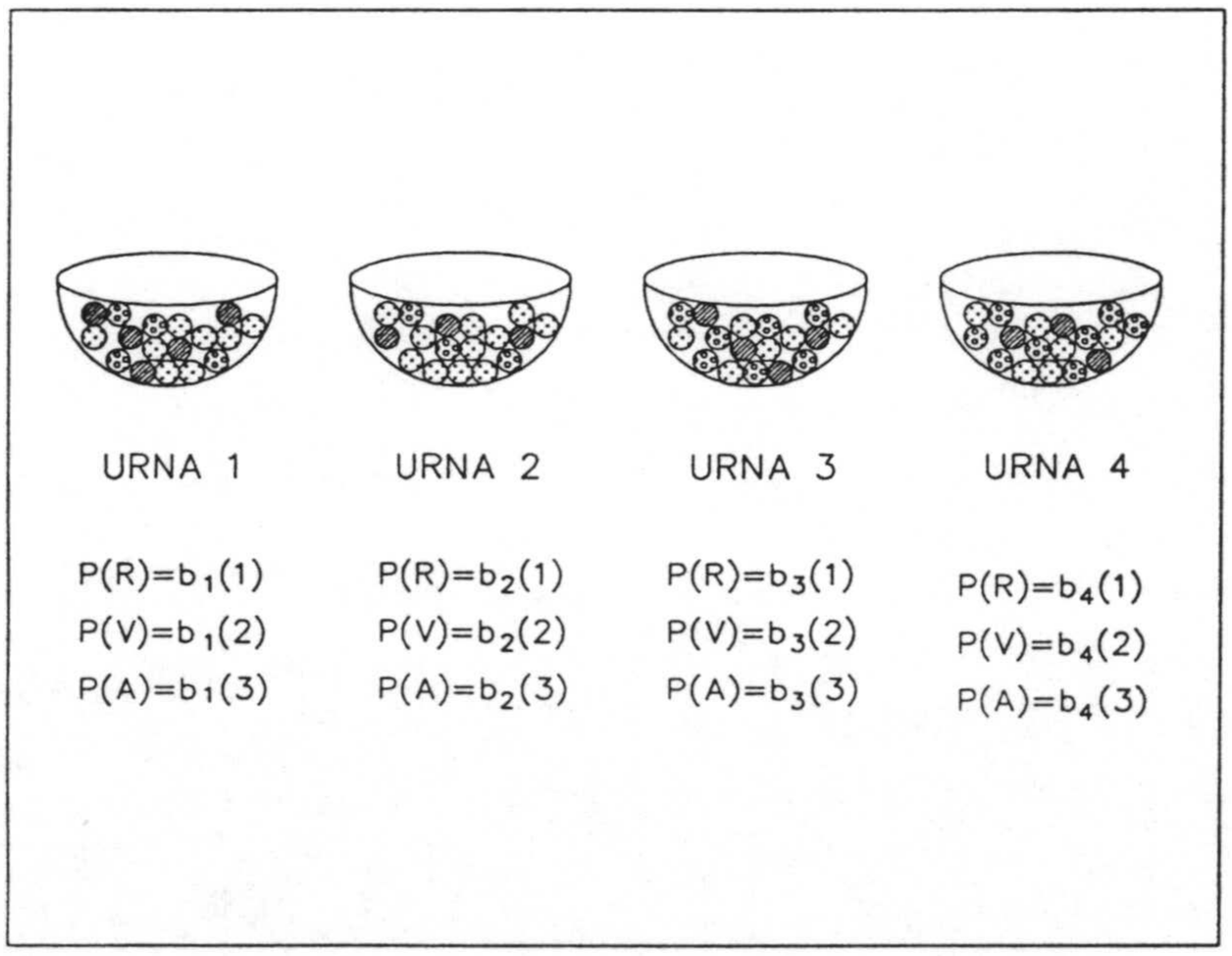

Figura 3.3. Ejemplo de las bolas y las urnas

que corresponderá a las observaciones del proceso, asociada a la cual existirá una secuencia de urnas seleccionadas durante el proceso de extracción

$$Y = \{U_1, U_4, U_3, U_1, U_2, \cdots, U_1\}$$

que permanece oculta al observador.

El proceso global puede caracterizarse por un proceso de Markov, que determina la selección de las sucesivas urnas, y que permanece oculto al observador, de forma que el estado del sistema queda caracterizado por la urna actualmente seleccionada; y un proceso mediante el cual se selecciona una bola a extraer de la urna actual, que es el observable, y que sólo depende de la urna actualmente seleccionada.

La caracterización de este proceso requiere la caracterización individual de los dos procesos que engloba, es decir, será necesario especificar el proceso de selección de urnas, que necesitará de las probabilidades de transición de estados, y de la distribución inicial de probabilidades de los estados del modelo; y la caracterización del proceso de extracción asociado a cada urna, determinado en este caso por las cantidades de bolas de cada color en cada urna, es decir, las probabilidades de extracción de cada tipo de bola en cada urna.

3.2.1. ELEMENTOS DE UN HMM

A continuación describiremos los elementos que determinan un modelo oculto de Markov, y la forma en que éste puede utilizarse para modelar un proceso estocástico como el descrito con anterioridad.

Un Modelo oculto de Markov queda caracterizado por lo siguiente:

1) *El conjunto de N estados del modelo.* Aunque la secuencia de estados permanece oculta al observador, es interesante en muchas situaciones, obtener información sobre la misma, ya que los estados suelen tener un significado físico (en el ejemplo, corresponden a la secuencia en que las urnas son seleccionadas). Denominaremos $S = \{s_i\}_{i=1..N}$ a este conjunto de estados, que supondremos finito.

2) *El conjunto de probabilidades de transición de estados.* Que denominaremos $A = \{a_{ij}\}_{i,j=1..N}$, definidas en la forma

$$a_{ij} = P(y_t=s_j \mid y_{t-1}=s_i) \qquad 1\leq i,j\leq N \tag{3.8}$$

donde y_t denota el estado del modelo en el instante de tiempo t. Este conjunto de probabilidades determinará la topología del modelo. Así, para un modelo en el que cada estado puede ser alcanzado desde cualquier otro en un sólo paso, $a_{ij}>0 \quad 1\leq i,j\leq N$. En general, los modelos pueden tener $a_{ij}=0$ para una o más parejas de valores (i,j). En cualquier caso deben verificar

$$\sum_{j=1}^{N} a_{ij} = 1 \qquad 1\leq i\leq N \tag{3.9a}$$

$$a_{ij} \geq 0 \qquad 1\leq i,j\leq N \tag{3.9b}$$

3) *La distribución de probabilidad de estados inciales.* Que denominaremos $\Pi=\{\pi_i\}_{i=1..N}$, definida en la forma

$$\pi_i = P(y_1=s_i) \qquad 1\leq i\leq N \tag{3.10}$$

donde y_1 representa el estado correspondiente al instante inicial.

4) *Las probabilidades de generación de observaciones.* Que caracterizarán el proceso de producción asociado a cada uno de los estados del modelo, y que denominaremos $B = \{b_i(v)\}_{i=1..N}$, definido en la forma

$$b_i(v) = P(x_t=v \mid y_t=s_i) \qquad 1\leq i\leq N \tag{3.11}$$

en donde x_t representa el valor de la observación en el instante de tiempo t. Supondremos que el proceso de generación de observaciones es independiente del tiempo, y que únicamente depende del estado actual del modelo. Estas probabilidades están representadas a través de funciones densidad de probabilidad en el caso en que las observaciones sean magnitudes contínuas, o distribuciones de probabilidad, en el caso en que las observaciones sean discretas. Ambos casos se discutirán más adelante, definiendo dos tipo de HMM, los discretos, o modelos con observaciones discretas y los contínuos, para los que las observaciones son contínuas.

De esta forma, un modelo HMM queda definido por la especificación de los conjuntos Π, A y B. En adelante utilizaremos la notación compacta

$$\lambda = (\Pi, A, B) \tag{3.12}$$

para referirnos a un determinado modelo λ. Una vez especificado, éste puede utilizarse para generar secuencias de observaciones $X = x_1 x_2 \cdots x_T$ en la forma siguiente:

1) Elegir un estado inicial $y_1 = s_i$ de acuerdo con la probabilidad π_i.

2) Fijar el contador de tiempo $t=1$.

3) Seleccionar una producción $x_t = v$ de acuerdo con la probabilidad $b_i(v)$.

4) Seleccionar un nuevo estado $y_{t+1} = s_j$ de acuerdo con la probabilidades a_{ij}.

5) Incrementar el contador de tiempo $t=t+1$. Si $t<T$ ir al paso 3, en otro caso terminar.

3.2.2. LOS TRES PROBLEMAS BASICOS DEL MODELADO HMM

Una vez definido el modelo para un determinado proceso, surgen tres problemas relacionados con la posible aplicación de dicho modelo, y que pasamos a describir a continuación.

Problema 1: Dada una secuencia de observaciones $X_1^T = x_1 x_2 \cdots x_T$ y un modelo λ, cómo evaluar eficientemente la probabilidad condicional $P(X_1^T \mid \lambda)$ de que la secuencia haya sido generada por el modelo.

Problema 2: Dada una secuencia de observaciones $X_1^T = x_1 x_2 \cdots x_T$ y un modelo λ, cómo determinar la secuencia de estados $Y_1^T = y_1 y_2 \cdots y_T$ que mejor explica (en un cierto sentido) la generación de la secuencia de observaciones por parte del modelo.

Problema 3: Dada una secuencia de observaciones $X_1^T = x_1 x_2 \cdots x_T$ correspondiente a un proceso que se desea modelar, cómo ajustar los parámetros del modelo $\lambda = (\Pi, A, B)$ de forma que se maximice la probabilidad de generación de dicha secuencia por el modelo.

El primero es un problema de evaluación que, una vez resuelto, nos permitirá evaluar eficientemente la probabilidad de generación de una cierta secuencia de observaciones por un modelo, probabilidad que se puede utilizar para clasificar las secuencias de observaciones. Punto básico de la aplicación de este tipo de modelos al reconocimiento.

La solución al segundo problema nos permitirá extraer información sobre el proceso oculto, en términos de la secuencia óptima de estados, lo que nos permitirá interpretar el significado de los estados del modelo, y extraer parámetros estadísticos sobre los estados del modelo como probabilidades de producción de observaciones, duraciones medias de los estados, etc. Estos parámetros se pueden incorporar al modelo como veremos más adelante.

El último es un problema de entrenamiento, cuya solución nos permitirá desarrollar un método para obtener los parámetros de un modelo en base a secuencias de observaciones del proceso que se va a modelar, de forma que se obtenga un modelo óptimo en el sentido de que maximice la probabilidad de generación de la secuencia de observaciones.

A continuación describiremos las soluciones a estos tres problemas para el caso particular de modelos con probabilidades de producción discretas; para extenderlas después a modelos con densidades de probabilidad de producción de observaciones contínuas.

3.3. MODELOS OCULTOS DE MARKOV DISCRETOS

Como ya indicamos en los apartados anteriores, en los HMM discretos se supone que las observaciones del proceso son símbolos pertenecientes a un alfabeto discreto y finito

$$V = \{v_i\} \qquad 1 \leq i \leq M \tag{3.13}$$

donde M es el número de símbolos del alfabeto.

Este tipo de modelos es para el que inicialmente se formularon las soluciones a los problemas de evaluación, entrenamiento y decodificación que a continuación pasamos a describir.

3.3.1. SOLUCION AL PROBLEMA DE EVALUACION

Supongamos una secuencia de observaciones (símbolos)

$$X_1^T = x_1 x_2 \cdots x_T \tag{3.14a}$$

de longitud T, y una secuencia de estados

$$Y_1^T = y_1 y_2 \cdots y_T \tag{3.14b}$$

donde y_1 es el estado[1] inicial de la secuencia.

La probabilidad de que un determinado modelo λ genere la secuencia de observaciones X_1^T cuando la secuencia de estados recorridos por el modelo es Y_1^T se puede escribir en la forma

$$P(X_1^T | Y_1^T, \lambda) = \prod_{t=1}^{T} P(x_t | y_t, \lambda) \tag{3.15}$$

donde se ha supuesto que las observaciones en diferentes instantes de tiempo son estadísticamente independientes.

De otro lado, la probabilidad de que el modelo recorra la secuencia de estados Y_1^T se puede escribir, utilizando la propiedad de Markov, en la forma

---

[1]En adelante supondremos que los estados están numerados del 1 al N, y que las variables y_t toman consecuentemente valores enteros entre 1 y N.

$$P(Y_1^T|\lambda) = P(y_T|Y_1^{T-1},\lambda)\cdot P(Y_1^{T-1}|\lambda)$$

$$= P(y_T|y_{T-1},\lambda)\cdot P(Y_1^{T-1}|\lambda)$$

$$= P(y_T|y_{T-1},\lambda)\cdot P(y_{T-1}|Y_1^{T-2},\lambda)\cdot P(Y_1^{T-2}|\lambda)$$

$$= P(y_T|y_{T-1},\lambda)\cdot P(y_{T-1}|y_{T-2},\lambda)\cdot P(Y_1^{T-2}|\lambda)$$

$$\cdots$$

$$= P(y_T|y_{T-1},\lambda)\cdot P(y_{T-1}|y_{T-2},\lambda) \cdots P(y_2|y_1,\lambda)\cdot P(y_1|\lambda)$$

$$= P(y_1|\lambda)\cdot\prod_{t=2}^{T} P(y_t|y_{t-1},\lambda) \tag{3.16a}$$

O bien, utilizando las definiciones de Π y A

$$P(Y_1^T|\lambda) = \pi_{y_1}\prod_{t=2}^{T} a_{y_{t-1}y_t} \tag{3.16b}$$

La probabilidad del suceso conjunto de la secuencia de observaciones y estados es

$$P(X_1^T,Y_1^T|\lambda) = P(X_1^T|Y_1^T,\lambda)\cdot P(Y_1^T|\lambda) \tag{3.17}$$

y por lo tanto, la probabilidad de generación de la secuencia de observaciones se puede obtener sumando la probabilidades conjuntas (3.17) para todas las posibles secuencias de estados

$$P(X_1^T|\lambda) = \sum_{Y_1^T} P(X_1^T|Y_1^T,\lambda)\cdot P(Y_1^T|\lambda) \tag{3.18a}$$

$$= \sum_{y_1 y_2 \cdots y_T} \pi_{y_1}\cdot b_{y_1}(x_1)\cdot\prod_{t=2}^{T} a_{y_{t-1}y_t}\cdot b_{y_t}(x_t) \tag{3.18b}$$

Esta expresión conlleva un número de cálculos del orden de $2TN^T$ productos. Basta con observar que, para cada uno de los sumandos de (3.18b), es necesario realizar $(2T-1)$ productos, y que la sumatoria múltiple contiene T sumatorias sencillas de N sumandos,

luego se tiene un total de N^T sumandos, y por lo tanto el número total de productos será de $(2T-1)N^T$.

Esta cantidad de productos hace inaplicable la expresión anterior para el cálculo de la probabilidad de generación ya que, incluso para valores moderados de número de estados y duraciones de secuencias, el número de productos es muy elevado (N=5 con T=50 resulta en 10^{37} productos y con T=100 en 10^{72} productos).

Afortunadamente, existe un algoritmo recursivo que permite obtener esta probabilidad de forma eficiente, es el denominado *Algoritmo Forward-Backward* [Baum68a, Baum67] (Adelante-Atrás en Español).

El algoritmo Forward-Backward

Consideremos la variable

$$\alpha_t(i) = P(x_1 x_2 \cdots x_t, y_t = s_i \mid \lambda) \tag{3.19}$$

que representa la probabilidad de generar la subsecuencia X_1^t, de forma que el modelo queda en el estado final s_i. Es fácil ver que para $\alpha_t(i)$ se puede establecer la siguiente recursión

Evaluación Forward

1) *Inicialización:*

$$\alpha_1(i) = \pi_i \cdot b_i(x_1) \qquad 1 \le i \le N \tag{3.20a}$$

2) *Recursión:*

$$\alpha_t(i) = \left[\sum_{j=1}^{N} \alpha_{t-1}(j) \cdot a_{ji}\right] \cdot b_i(x_t) \qquad \begin{array}{l} t=1,2,\cdots,T \\ 1 \le i \le N \end{array} \tag{3.20b}$$

3) *Terminación:*

$$P(X_1^T \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{3.20c}$$

El proceso de cálculo descrito por la recursión previa es el siguiente. En el paso de inicialización, sólo se evalua la probabilidad de generar el primer símbolo en uno de los N estados del modelo. La recursión establece que la probabilidad de generar una subcadena de t símbolos terminando en un cierto estado s_i tiene en cuenta la contribución de todas las posibles secuencias de estados que generan la subcadena de $t-1$ símbolos previos finalizando en un cierto estado s_j multiplicadas por las probabilidades de transición entre estos estados y el s_i, y por último, la probabilidad de generar el último símbolo x_t de la subcadena en el estado final s_i.

En la figura 3.4 se puede ver esquematizado el proceso de cálculo necesario para la evaluación de la probabilidad de generación con este algoritmo. En la figura 3.4a se muestran las operaciones necesarias para un paso de la recursión en el cálculo de las probabilidades parciales $\alpha_t(i)$, y en la 3.4b se muestra la red de operaciones necesarias para la evaluación de la probabilidad de generación de una cadena de 4 símbolos para un modelo con 3 estados.

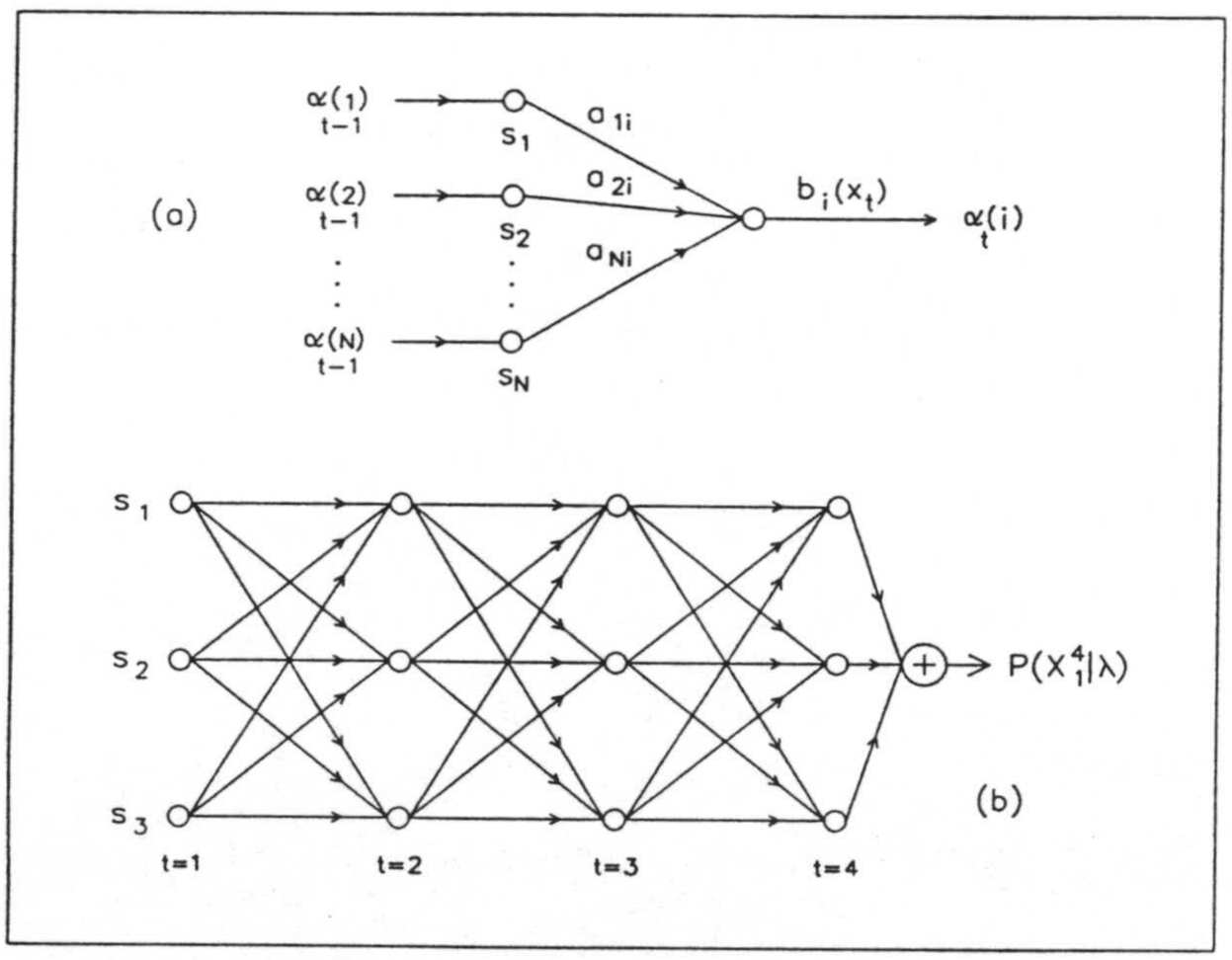

Figura 3.4. Evaluación Forward

De la recursión expuesta para el cálculo de $\alpha_t(i)$, se puede deducir fácilmente el número de productos necesarios para la evaluación de la probabilidad de generación (o equivalentemente $\alpha_T(i)$ $1\leq i\leq N$). Para el cálculo $\alpha_t(i)$ se requieren $(N+1)$ productos para los valores $2\leq t\leq T$, y un producto para $t=1$; como hay que calcular N valores para cada instante de tiempo, tendremos en total $N((N+1)(T-1)+1)$ productos, es decir que la complejidad es $O(TN^2)$ productos frente a los $O(TN^T)$ requeridos para la evaluación directa de (3.18b).

Aunque no es necesaria la segunda parte del algoritmo para la evaluación de las probabilidades de generación de cadenas, si lo será en la solución del problema de entrenamiento y por lo tanto pasamos a describirla a continuación.

Supongamos una variable definida en la forma

$$\beta_t(i) = P(x_{t+1}x_{t+2}\cdots x_T \,|\, y_t=s_i,\lambda) \tag{3.21}$$

que representa la probabilidad de que el modelo genere una subcadena partiendo del estado $y_t=s_i$ hasta el final $t=T$.

Con esta definición, se puede establecer una recursión de la forma

Evaluación Backward

1) *Inicialización:*

$$\beta_t(i) = 1 \qquad 1\leq i\leq N \tag{3.22a}$$

2) *Recursión:*

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}\cdot b_j(x_{t+1})\cdot\beta_{t+1}(j) \qquad \begin{array}{l} t=T-1,T-2,\cdots,1 \\ 1\leq i\leq N \end{array} \tag{3.22b}$$

3) *Terminación:*

$$P(X_1^T|\lambda) = \sum_{i=1}^{N} \pi_i\cdot b_i(x_1)\cdot\beta_1(i) \tag{3.22c}$$

De la recursión (3.22) se puede deducir de forma sencilla, que la complejidad computacional es, como en el caso de la evaluación *Forward*, de $O(N^2T)$ productos.

En adelante denominaremos *probabilidades adelante* a las cantidades $\alpha_t(i)$ y *probabilidades hacia atrás* a las $\beta_t(i)$. En la figura 3.5a se esquematizan las operaciones necesarias para el cálculo recursivo de las probabilidades hacia atrás.

3.3.2. SOLUCION AL PROBLEMA DE DECODIFICACION

A diferencia del problema de evaluación, no existe una solución única al problema de la obtención de la secuencia óptima de estados dada una secuencia de observaciones, sino que depende del criterio con que se defina esta secuencia óptima, así, un posible criterio es el de extraer la secuencia de estados que verifica que las probabilidades de cada uno de los estados que la componen es máxima (dada la secuencia de observaciones y el modelo). Esta se puede obtener sin más que tener en cuenta que la magnitud

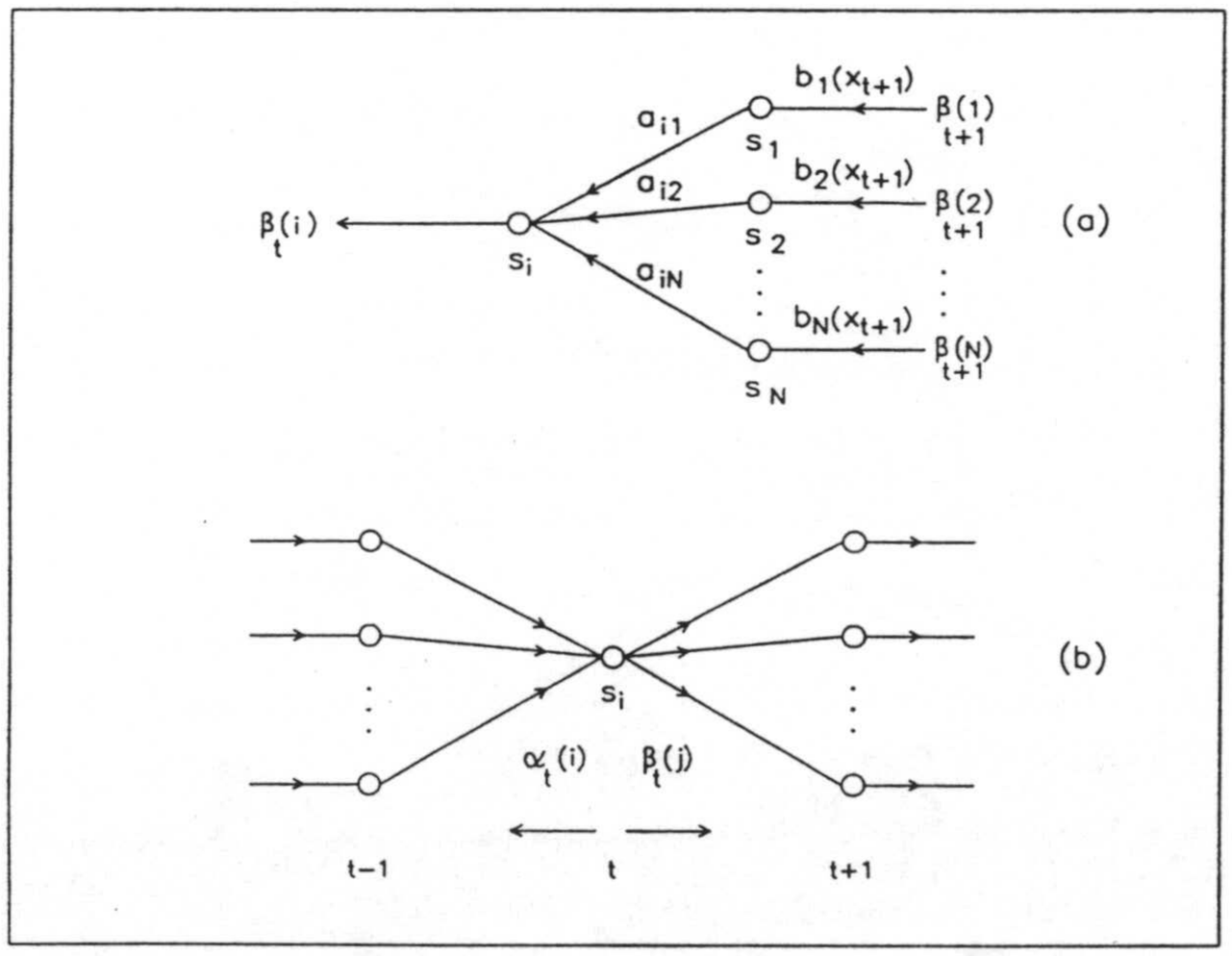

Figura 3.5. a) Evaluación Backward y b) probabilidades adelante y atrás

$$\gamma_t(i) = P(y_t = s_i \,|X_1^T, \lambda) \tag{3.23}$$

que representa la probabilidad de que el modelo λ se encuentre en el estado s_i en el tiempo t, durante la generación de la cadena de símbolos X_1^T, puede calcularse en función de las probabilidades adelante y atrás en la forma

$$\gamma_t(i) = \frac{\alpha_t(i)\,\beta_t(i)}{P(X_1^T|\lambda)} \tag{3.24a}$$

$$= \frac{\alpha_t(i)\,\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\,\beta_t(j)} \qquad \begin{matrix} 1 \le i \le N \\ 1 \le t \le T \end{matrix} \tag{3.24b}$$

dado que el producto de las probabilidades adelante y atrás para un mismo estado s_i en el mismo instante de tiempo t representa la probabilidad de que el modelo visite el estado s_i en el instante de tiempo t durante la generación de una determinada cadena de observaciones (ver figura 3.4), mientras que la suma sobre los N estados del modelo representa la probabilidad total de generación de la secuencia de observaciones. Este valor se denomina usualmente *Probabilidad de ocupación de estados*.

Con esta definición, para extraer la secuencia de estados de máxima probabilidad basta con seleccionar aquellos estados para los que $\gamma_t(i)$ es máxima

$$Y_1^{T*} = y_1^* y_2^* \cdots y_T^* \tag{3.25a}$$

$$y_t^* = \underset{1 \le i \le N}{\operatorname{argmax}} \{\gamma_t(i)\} \qquad 1 \le t \le T \tag{3.25b}$$

Ese método, aunque sencillo, presenta varios inconvenientes. El primero de éstos es que, al realizar una optimización local de estados, es posible obtener una secuencia de estados imposible para el modelo si éste contiene transiciones prohibidas (algunos a_{ij} son nulos). Además, no está garantizado que la secuencia de estados obtenida sea la de máxima probabilidad de generación de la secuencia de símbolos. Por estos motivos, se suele utilizar un criterio diferente en la selección de la secuencia óptima de estados que es el siguiente: Seleccionar la secuencia de estados para la que la probabilidad de generación condicionada es máxima

$$Y_1^{T*} = \underset{Y_1^T}{\text{argmax}} \{P(X_1^T | Y_1^T, \lambda\} \tag{3.26}$$

De nuevo, el cálculo directo de (3.26) para la obtención de la secuencia óptima de estados presenta una complejidad que hace inaplicable dicha relación incluso para valores razonables de N y T. En su lugar, expondremos un algoritmo recursivo del estilo del *Forward-Backward* denominado *Algoritmo de Viterbi* [Viterbi67a, Forney73]

Algoritmo de Viterbi

Definamos la variable

$$\delta_t(i) = P^*(x_1 x_2 \cdots x_T | y_t = s_i, \lambda) \tag{3.27a}$$

$$= \max_{y_1 y_2 \cdots y_{t-1}} \{P(x_1 x_2 \cdots x_t | y_1 y_2 \cdots y_{t-1}, y_t = s_i, \lambda)\}\} \tag{3.27b}$$

Cuyo significado no es otro que el de la probabilidad máxima de generación de una secuencia de t símbolos sobre cualquier secuencia simple de estados cuyo estado final es el s_i. Es sencillo demostrar que esta variable verifica una relación de recurrencia de la forma

$$\delta_t(i) = \max_{1 \le j \le N} \{\delta_{t-1}(j) \cdot a_{ji}\} \cdot b_i(x_t) \tag{3.28}$$

Es claro que, para recuperar la secuencia de estados que generan la probabilidad máxima, es necesario almacenar los valores del argumento que maximizan (3.28), y para ésto utilizamos una matriz $\psi_t(i)$. A continuación se presenta el algoritmo completo para la obtención de la probabilidad máxima de generación, y la secuencia correspondiente de estados.

Algoritmo de Viterbi

1) *Inicialización:*

$$\delta_t(i) = \pi_i \cdot b_i(x_1) \qquad 1 \le i \le N \qquad (3.29a)$$

2) *Recursión:*

$$\delta_t(i) = \max_{1 \le j \le N} \{\delta_{t-1}(j) \cdot a_{ji}\} \cdot b_i(x_t) \qquad \begin{matrix} t=1,2,\cdots,T \\ 1 \le i \le N \end{matrix} \qquad (3.29b)$$

$$\psi_t(i) = \operatorname*{argmax}_{1 \le j \le N} \{\delta_{t-1}(j) \cdot a_{ji}\} \qquad \begin{matrix} t=1,2,\cdots,T \\ 1 \le i \le N \end{matrix} \qquad (3.29c)$$

3) *Terminación:*

$$P^*(X_1^T|\lambda) = \max_{1 \le i \le N} \{\delta_T(i)\} \qquad (3.29d)$$

$$y_T^* = \operatorname*{argmax}_{1 \le i \le N} \{\delta_T(i)\} \qquad (3.29e)$$

4) *Recursión para obtener la secuencia de estados:*

$$y_t^* = \psi_{t+1}(y_{t+1}^*) \qquad t=T-1,T-2,\cdots,1 \qquad (3.29f)$$

3.3.3. SOLUCION AL PROBLEMA DE ENTRENAMIENTO

Este es el más complejo de los tres problemas relacionados con el modelado HMM, el de seleccionar el conjunto de parámetros del modelo $\lambda(\Pi,A,B)$ que maximiza la probabilidad de generación de una determinada secuencia de observaciones X_1^T. No existe ningún método analítico conocido para resolver el problema, es más, dada una secuencia finita de observaciones, no es posible estimar de forma óptima los parámetros del modelo [Rabiner89]. Sin embargo, se pueden optimizar los parámetros del modelo de forma que se maximice localmente la función probabilidad $P(X_1^T|\lambda)$ mediante métodos iterativos como el procedimiento *Baum-Welch* (o equivalentemente un método EM de expectación-modificación [Dempster77]), o mediante técnicas de descenso por gradiente [Levinson83].

A continuación expondremos el método de reestimación Baum-Welch, desarrollado inicialmente por Baum [Baker75a, Baum68], que garantiza la convergencia uniforme hacia un máximo local de la función probabilidad de generación.

En una forma similar a como se definición la probabilidad de ocupación de estados (3.23), se puede definir la probabilidad de transición de estados

$$\xi_t(i,j) = P(y_t=s_i, y_{t+1}=s_j \mid X_1^T, \lambda) \tag{3.30}$$

que representa la probabilidad de que el modelo se encuentre en el estado s_i en el instante de tiempo t, y se produzca una transición de forma que en el instante de tiempo $t+1$, el estado sea el s_j. Este valor puede expresarse, en términos de las probabilidades adelante y atrás, en la forma

$$\xi_t(i,j) = \frac{\alpha_t(i)\, a_{ij}\, b_j(x_{t+1})\, \beta_{t+1}(j)}{P(X_1^T \mid \lambda)} \tag{3.31a}$$

$$= \frac{\alpha_t(i)\, a_{ij}\, b_j(x_{t+1})\, \beta_{t+1}(j)}{\sum_{k=1}^{N}\sum_{l=1}^{N} \alpha_t(k)\, a_{kl}\, b_l(x_{t+1})\, \beta_{t+1}(l)} \qquad \begin{matrix} 1\le i\le N \\ 1\le j\le N \\ 1\le t\le T-1 \end{matrix} \tag{3.31b}$$

en la figura 3.6 se muestra esquemáticamente la situación necesaria para el cálculo de $\xi_t(i,j)$.

Sumando sobre el índice de tiempo los valores de probabilidad de ocupación de estados, y de transición, se pueden obtener los valores esperados (para una cadena de símbolos y un modelo dados) del número de veces que un estado es visitado y el número de transiciones que se producen entre dos estados del modelo.

Si sumamos $\gamma_t(i)$ para $1\le t\le T$, obtendremos el número esperado de veces que el modelo se encuentra en el estado s_i, y si sumamos para $1\le t\le T-1$, obtendremos el número esperado de veces en que el modelo realiza una transición desde el estado s_i. Además, si sumamos para $1\le t\le T$, con la restricción de que el símbolo observado sea el v_k, obtendremos el número esperado de veces que el modelo genera el símbolo v_k en el estado s_i. Por último, sumando $\xi_t(i,j)$ para $1\le t\le T-1$, obtendremos el número esperado de veces

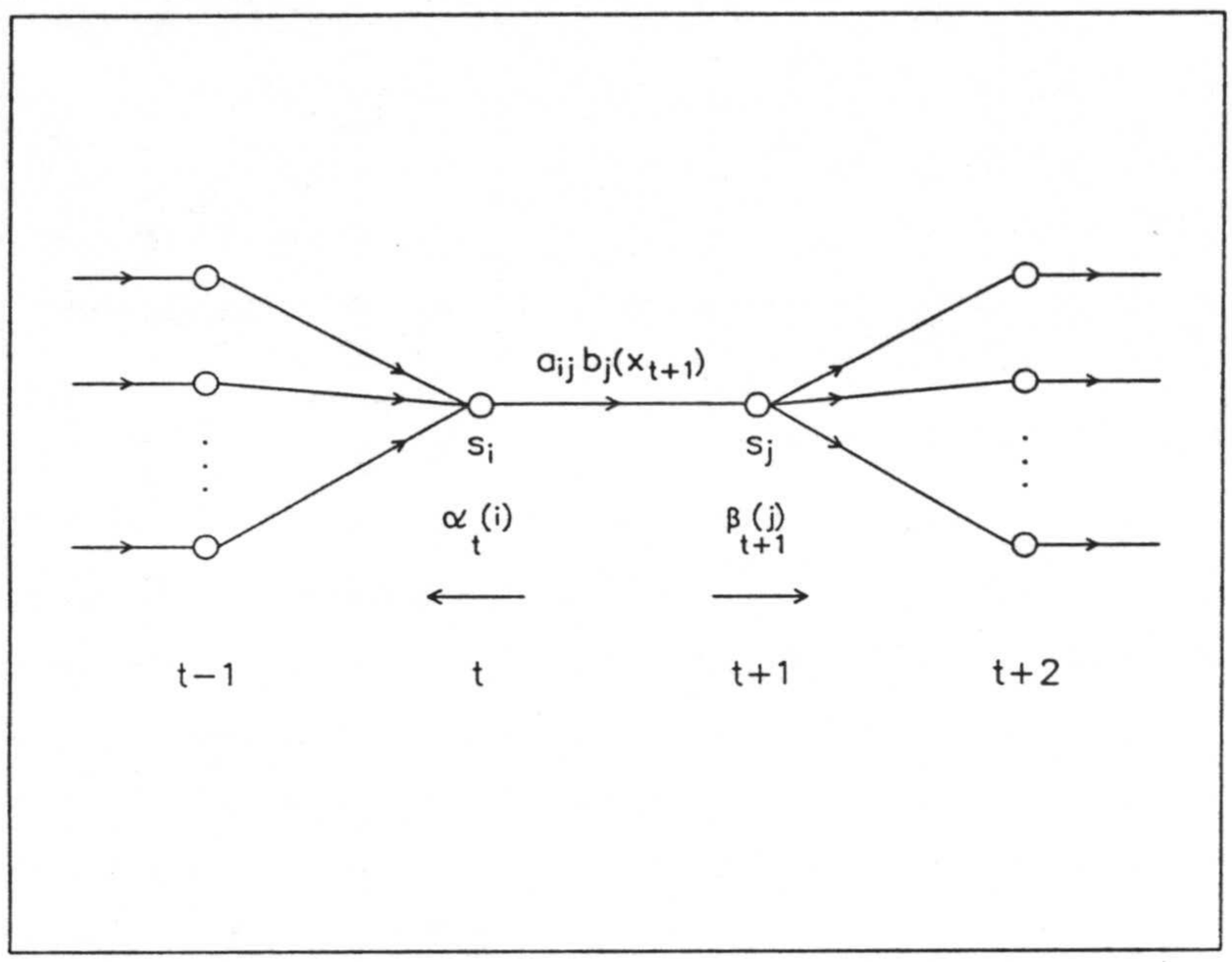

Figura 3.6. Esquema para el cálculo de $\xi_t(i,j)$

que se produce un transición entre los estados s_i y s_j.

$$\sum_{t=1}^{T} \gamma_t(i) = \text{número esperado de visitas a } s_i \tag{3.32a}$$

$$\sum_{\substack{t=1 \\ x_t=v_k}}^{T} \gamma_t(i) = \text{número esperado de observaciones } v_k \text{ en } s_i \tag{3.32b}$$

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{número esperado de transiciones desde } s_i \tag{3.32c}$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{número esperado de transiciones desde } s_i \text{ a } s_j \tag{3.32d}$$

Con estas definiciones, se puede establecer las siguientes fórmulas para la reestimación de los parámetros obteniendo un nuevo modelo $\bar{\lambda}(\bar{\Pi},\bar{A},\bar{B})$

$$\bar{\pi}_i = \gamma_1(i) \qquad 1 \le i \le N \tag{3.33a}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \qquad \begin{matrix} 1 \le i \le N \\ 1 \le j \le N \end{matrix} \tag{3.33b}$$

$$\bar{b}_i(k) = \frac{\sum_{\substack{t=1 \\ x_t = v_k}}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)} \qquad \begin{matrix} 1 \le i \le N \\ 1 \le k \le M \end{matrix} \tag{3.33c}$$

Baum [Baum68] mostró que si, a partir de un modelo $\lambda(\Pi, A, B)$, y utilizando las ecuaciones (3.33), se obtiene un nuevo modelo $\bar{\lambda}(\bar{\Pi}, \bar{A}, \bar{B})$, la probabilidad de generación de la secuencia dado el modelo $\bar{\lambda}$ es siempre mayor que la obtenida para el modelo inicial λ, excepto cuando se alcanza un valor crítico de la función probabilidad en cuyo caso las dos probabilidades coinciden.

Esta prueba garantiza una convergencia uniforme del método de reestimación, y es la base para el desarrollo de un algoritmo iterativo de reestimación de los parámetros del modelo, simplemente estimando y actualizando los parámetros del modelo hasta un punto a partir del cual la probabilidad no varía apreciablemente. Este método es comunmente denominado *Estimación de Máxima Probabilidad* [Dempster77], dado que maximiza la función probabilidad de generación en función de los parámetros del modelo.

Otra ventaja de este método es que las restricciones estocásticas sobre los parámetros del modelo reestimado se verifican de forma automática. Es fácil de comprobar a partir de (3.33) y las definiciones de $\gamma_t(i)$ y de $\xi_t(i,j)$, que se verifica

$$\sum_{i=1}^{N} \bar{\pi}_i = 1 \tag{3.34a}$$

$$\sum_{j=1}^{N} \bar{a}_{ij} = 1 \qquad 1 \le i \le N \tag{3.34b}$$

$$\sum_{k=1}^{M} \bar{b}_i(k) = 1 \qquad 1 \le i \le N \tag{3.34c}$$

Las ecuaciones (3.33) pueden obtenerse de forma directa maximizando la función auxiliar de Baum

$$Q(\lambda,\bar{\lambda}) = \sum_{Y_1^T} P(X_1^T, Y_1^T \mid \lambda) \cdot \log P(X_1^T, Y_1^T \mid \bar{\lambda}) \tag{3.35}$$

para la que es sencillo comprobar que se verifica

$$Q(\lambda,\bar{\lambda}) - Q(\lambda,\lambda) \le P(X_1^T \mid \bar{\lambda}) - P(X_1^T \mid \lambda) \tag{3.36}$$

y consecuentemente

$$Q(\lambda,\bar{\lambda}) \ge Q(\lambda,\lambda) \quad \Rightarrow \quad P(X_1^T \mid \bar{\lambda}) \ge P(X_1^T \mid \lambda) \tag{3.37}$$

Alternativamente, se pueden obtener las relaciones (3.33) maximizando directamente las función probabilidad $P = P(X_1^T \mid \lambda)$ mediante un método tradicional tal como el de los multiplicadores de Lagrange, para el que se obtienen las relaciones

$$\bar{\pi}_i = \frac{\pi_i \frac{\partial P}{\partial \pi_i}}{\sum_{k=1}^{N} \pi_k \frac{\partial P}{\partial \pi_k}} \qquad 1 \le i \le N \tag{3.38a}$$

$$\bar{a}_{ij} = \frac{a_{ij} \frac{\partial P}{\partial a_{ij}}}{\sum_{k=1}^{N} a_{ik} \frac{\partial P}{\partial a_{ik}}} \qquad \begin{matrix} 1 \le i \le N \\ 1 \le j \le N \end{matrix} \tag{3.38b}$$

$$\bar{b}_i(k) = \frac{b_i(k) \frac{\partial P}{\partial b_i(k)}}{\sum_{l=1}^{M} b_i(l) \frac{\partial P}{\partial b_i(l)}} \qquad \begin{matrix} 1 \le i \le N \\ 1 \le k \le M \end{matrix} \tag{3.38c}$$

a partir de la cuales, expresando adecuadamente $P(X_1^T|\lambda)$ en función de las probabilidades adelante y atrás, se obtienen las mismas relaciones que en (3.33).

Por último, indicar que el problema puede contemplarse globalmente como una optimización de parámetros, y utilizar técnicas de descenso por gradiente para la estimación iterativa de los parámetros del modelo. Esta aproximación permite utilizar otros criterios de optimización como el MMI (o de Máxima Información Mútua) [Lowerre80] que se basa en maximizar la información mútua promedio entre el conjunto de secuencias de entrenamiento y el conjunto de modelos a diseñar, cuando se desea diseñar conjuntamente una serie de modelos que se utilizarán con propósitos discriminativos.

3.4. MODELOS OCULTOS DE MARKOV CONTINUOS

Como ya indicamos en el apartado 3.2.1, dentro de la formulación general de los modelos ocultos de Markov, puede suponerse que la variable estocástica asociada a la producción de observaciones es contínua y en general multivariada, lo que fuerza a modelar las probabilidades de producción de observaciones como funciones densidad de probabilidad contínuas de dicha variable. Asumido ésto, es necesario seleccionar una forma paramétrica para las funciones densidad de probabilidad que caracterizan las observaciones en cada estado del modelo, y establecer fórmulas para la estimación de los parámetros de dichas funciones densidad de probabilidad. Las fórmulas de estimación del resto de los

parámetros del modelo, así como los algoritmos de evaulación y decodificación antes expuestos, no se ven afectados por esta modificación más que en lo que concierne a la evaluación de los valores $b_i(x_t)$.

La forma más general para las funciones densidad de probabilidad asociadas a los estados del modelo, para las que se han descrito fórmulas explícitas de reestimación [Liporace82a, Juang86], es la formada por una combinación lineal finita de la forma

$$b_i(x) = \sum_{m=1}^{M} c_{im} \cdot \Theta(x, \mu_{im}, \Sigma_{im}) \qquad 1 \le i \le N \tag{3.39}$$

donde x es un vector contínuo de observaciones (generalmente multivariado), y Θ es una función densidad de probabilidad gausiana de vector media μ_{im} y matriz de covarianza Σ_{im} que forma parte de la combinación lineal con peso relativo c_{im}, valores que denominaresmos en adelante *coeficientes de la mezcla*. Estos coeficientes verifican las relaciones

$$\sum_{m=1}^{M} c_{im} = 1 \qquad 1 \le i \le N \tag{3.40a}$$

$$c_{im} \ge 0 \qquad \begin{matrix} 1 \le i \le N \\ 1 \le m \le M \end{matrix} \tag{3.40b}$$

de forma que las densidades de probabilidad así obtenidas verifican la propiedad de normalización

$$\int_{-\infty}^{+\infty} b_i(x)\, dx = 1 \qquad 1 \le i \le N \tag{3.41}$$

Esta formulación es general, dado que una función densidad de probabilidad como la descrita en (3.39), puede aproximar arbitrariamente cualquier a cualquier densidad de probabilidad finita sin más que tomar un número suficientemente elevado de términos en la combinación lineal.

Se puede demostrar [Liporace82a, Juang86] que las fórmulas de reestimación para los coeficientes de la mezcla así como para los vectores media y matrices de covarianza son de la forma siguiente

$$\bar{c}_{im} = \frac{\sum_{t=1}^{T} \gamma_t(i,m)}{\sum_{k=1}^{M} \sum_{t=1}^{T} \gamma_t(i,k)} \qquad \begin{matrix} 1 \le i \le N \\ 1 \le m \le M \end{matrix} \qquad (3.42a)$$

$$\bar{\mu}_{im} = \frac{\sum_{t=1}^{T} \gamma_t(i,m) \cdot x_t}{\sum_{k=1}^{M} \sum_{t=1}^{T} \gamma_t(i,k)} \qquad \begin{matrix} 1 \le i \le N \\ 1 \le m \le M \end{matrix} \qquad (3.42b)$$

$$\bar{\Sigma}_{im} = \frac{\sum_{t=1}^{T} \gamma_t(i,m) \cdot (x_t - \mu_{im}) \cdot (x_t - \mu_{im})^T}{\sum_{k=1}^{M} \sum_{t=1}^{T} \gamma_t(i,k)} \qquad \begin{matrix} 1 \le i \le N \\ 1 \le m \le M \end{matrix} \qquad (3.42c)$$

Siendo $\gamma_t(i,m)$ un generalización de la variable $\gamma_t(i)$ definida para la reestimación en modelos discretos, que representa la probabilidad de que el modelo utilice el componente m de la mezcla en el estado s_i y en el tiempo t para la generación de la observación

$$\gamma_t(i,m) = \left[\frac{\alpha_t(i) \cdot \beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j) \cdot \beta_t(j)} \right] \cdot \left[\frac{c_{im} \cdot \Theta(x_t, \mu_{im}, \Sigma_{im})}{\sum_{k=1}^{M} c_{ik} \cdot \Theta(x_t, \mu_{ik}, \Sigma_{ik})} \right] \qquad \begin{matrix} 1 \le i \le N \\ 1 \le m \le M \end{matrix} \qquad (3.43)$$

Aunque esta formulación es general, se pueden encontrar en la bibliografía [Poritz82a, Juang85] formulaciones particulares para modelos autorregresivos de las observaciones, que no discutiremos aquí ya que no utilizaremos este modelado para las observaciones.

3.5. IMPLEMENTACION DE LOS MODELOS

Como parte final de este capítulo, discutiremos algunas cuestiones relacionadas con la implementación del modelado HMM. En concreto trataremos las referentes al rango de valores de las probabilidades de generación, la estimación de los parámetros del modelo con mútiples secuencias de observaciones, la elección de los valores iniciales de los parámetros del modelo, y los efectos del número finito de observaciones.

3.5.1. ESCALADO Y COMPRESION LOGARITMICA

Si consideramos la expresión (3.19), que define la probabilidad hacia delante $\alpha_t(i)$ y (3.18a-b), es fácil concluir que ésta puede expresarse en la forma

$$\alpha_t(i) = \sum_{y_1 y_2 \cdots y_{t-1}} \left[\pi_{y_1} \cdot \prod_{\tau=2}^{t-1} a_{y_{\tau-1} y_\tau} \right] \cdot \left[\prod_{\tau=1}^{t-1} b_{y_\tau}(x_\tau) \right] \tag{3.44}$$

y dado que π_i, a_{ij} y $b_i(x)$ son, en valor absoluto, inferiores a la unidad (frecuentemente muy inferiores), el valor absoluto de $\alpha_t(i)$ decae exponencialmente a cero con t, lo que presenta un problema de representación para éstos valores cuando los algoritmos se han de implementar en una computadora. Se han propuesto al menos dos soluciones a este problema, que pasamos a describir a continuación.

Compresión Logarítmica

Una posibilidad es la de representar las probabilidades a través de sus logaritmos, [Lee88b, Brown87], lo que realiza una compresión logarítmica de su rango de valores. Esta aproximación tiene el inconveniente de que es necesario calcular el logaritmo de una suma de probabilidades en función de los logaritmos de las mismas. No existe solución analítica exacta a este problema, sin embargo se puede obtener una solución aproximada de la forma siguiente:

Supongamos que P_1 y P_2 son dos probabilidades tales que $P_1 \geq P_2$. Podemos escribir

$$\log_b (P_1+P_2) = \log_b \left[b^{\log_b P_1} + b^{\log_b P_2} \right] \tag{3.45a}$$

$$= \log_b \left[b^{\log_b P_1} . \left[1+b^{(\log_b P_2 - \log_b P_1)} \right] \right] \tag{3.45b}$$

$$= \log_b P_1 + \log_b \left[1+b^{(\log_b P_2 - \log_b P_1)} \right] \tag{3.45c}$$

de la misma forma se puede escribir una expresión simétrica para el caso $P_2 \geq P_1$, de forma que el exponente de b en (3.45c) sea menor que la unidad, y establecer una tabla de valores de la forma

$$T(n) = \begin{cases} \log_b (1+b^n) & ; \ T(n) \geq 0.5 \\ 0 & ; \text{ en otro caso} \end{cases} \tag{3.46}$$

con $n = 0,-1,-2, \cdots ,-L$, de donde puede obtenerse al valor de L en función de la base b del logaritmo en la forma

$$L \geq -\log_b (\sqrt{b} - 1) \tag{3.47}$$

La elección de la base b determina la precisión de la representación. Así, K.F. Lee [Lee88b] utilizó una base b=1.0001, con una tabla de 99041 valores. De esta forma, se puede utilizar la aproximación

$$\log_b (P_1,P_2) = \begin{cases} \log_b P_1 + T\,[log_b P_2 - \log_b P_1] & ; \ P_1 \geq P_2 \\ \log_b P_2 + T\,[log_b P_1 - \log_b P_2] & ; \ P_2 \geq P_1 \end{cases} \tag{3.48}$$

Escalado dinámico

Una solución exacta al problema de la representación de las probabilidades de generación del modelo es la de modificar los algoritmos de cálculo para introducir un escalado dinámico de los valores [Levinson83] en función del índice de la recursión t.

Para hacer ésto definiremos, para el algoritmo *Forward-Bakward*, la variable escalada $\hat{\alpha}_t(i)$, y la recursión modificada

$$\tilde{\alpha}_t(i) = \left[\sum_{j=1}^{N} \hat{\alpha}_t(j)\cdot a_{ji}\right]\cdot b_j(x_t) \tag{3.49a}$$

$$c_t = \frac{1}{\sum_{j=1}^{N} \tilde{\alpha}_t(j)} \tag{3.49b}$$

$$\hat{\alpha}_t(i) = c_t \cdot \tilde{\alpha}_t(i) \tag{3.49c}$$

Según esta recursión puede comprobarse [Rabiner89] que se verifican las relaciones

$$\hat{\alpha}_t(i) = \left[\prod_{\tau=1}^{t} c_\tau\right]\cdot\alpha_t(i) = C_t\cdot\alpha_t(i) \tag{3.50a}$$

$$= \frac{\alpha_t(i)}{\sum_{j=1}^{N} \alpha_t(j)} \tag{3.50b}$$

$$= \frac{\alpha_t(i)}{P(X_1^t|\lambda)} \qquad \begin{matrix} 1\le i\le N \\ 1\le t\le T \end{matrix} \tag{3.50c}$$

Con lo que los valores escalados $\hat{\alpha}_t(i)$ son comparables para cualquier valor de t ya que el escalado fuerza a que estos valores sean los de $\alpha_t(i)$ relativos a la probabilidad total de generación de la subcadena X_1^t.

Dado que de (3.50) se tiene

$$P(X_1^t|\lambda) = \frac{1}{C_t} = \frac{1}{\prod_{\tau=1}^{t} c_\tau} \tag{3.51}$$

el logartimo de la probabilidad de generación puede calcularse en la forma

$$\log P(X_1^T|\lambda) = -\sum_{t=1}^{T} \log c_t \tag{3.52}$$

Con estas definiciones, la recursión *Forward* queda modificada de la siguiente forma

Evaluación Forward escalada

1) *Inicialización:*

$$\tilde{\alpha}_1(i) = \pi_i \cdot b_i(x_1) \tag{3.53a}$$

$$c_1 = \frac{1}{\sum_{i=1}^{N} \tilde{\alpha}_t(i)}$$

$$\hat{\alpha}_t(i) = c_1 \cdot \tilde{\alpha}_1(i)$$

$$K_1 = \log c_1$$

2) *Recursión:*

$$\tilde{\alpha}_t(i) = \left[\sum_{j=1}^{N} \hat{\alpha}_{t-1}(j) \cdot a_{ji}\right] \cdot b_i(x_t) \tag{3.53b}$$

$$c_t = \frac{1}{\sum_{i=1}^{N} \tilde{\alpha}_t(i)}$$

$$\hat{\alpha}_t(i) = c_t \cdot \tilde{\alpha}_t(i)$$

$$K_t = K_{t-1} + \log c_t$$

3) *Terminación:*

$$\log P(X_1^T | \lambda) = -K_T \tag{3.53c}$$

Con respecto a la recursión para obtener las probabilidades hacia atrás $\beta_t(i)$, se podría desarrollar un algoritmo de escalado similar al anteriormente descrito, pero dado que los valores de estas probabilidades son de magnitud comparable a la de las probabilidades hacia delante, y para evitar cálculo innecesario, se pueden simplemente utilizar los valores c_t para realizar el escalado, quedando el algoritmo modificado en la forma

Evaluación Backward escalada

1) *Inicialización:*

$$\tilde{\beta}_T(i) = c_T \tag{3.54a}$$

2) *Recursión:*

$$\tilde{\beta}_t(i) = \sum_{j=1}^{N} a_{ij} \cdot b_j(x_{t+1}) \cdot \hat{\beta}_{t+1}(j) \tag{3.54b}$$

$$\hat{\beta}_t(i) = c_t \cdot \tilde{\beta}_t(i)$$

En base a la definición de estos algoritmos, es sencillo obtener las relaciones entre las probabilidades y sus correspondientes escaladas en la forma

$$\hat{\alpha}_t(i) = \left[\prod_{\tau=1}^{t} c_\tau\right] \cdot \alpha_t(i) = C_t \cdot \alpha_t(i) \tag{3.55a}$$

$$\hat{\beta}_t(i) = \left[\prod_{\tau=t}^{T} c_\tau\right] \cdot \beta_t(i) = D_t \cdot \beta_t(i) \tag{3.55b}$$

Verificándose además

$$C_t \cdot D_{t+1} = \left[\prod_{\tau=1}^{T} c_\tau\right] = \frac{1}{P(X_1^T|\lambda)} \tag{3.56}$$

Modificaciones a las fórmulas de reestimación

Las fórmulas de reestimación no se ven afectadas por el proceso de escalado más que en lo que concierne al cálculo de los valores de las probabilidades de ocupación de estados $\gamma_t(i)$ y de transición de estados $\xi_t(i,j)$. Estas se pueden calcular a partir de las probabilidades hacia delante y atrás escaladas en la forma

$$\xi_t(i,j) = \frac{\alpha_t(i)\cdot a_{ij}\cdot b_i(x_{t+1})\cdot \beta_j(t+1)}{P(X_1^T|\lambda)} \tag{3.57a}$$

$$= C_t\cdot\alpha_t(i)\cdot a_{ij}\cdot b_i(x_{t+1})\cdot\beta_j(t+1)\cdot D_{t+1} \tag{3.57b}$$

$$= \hat{\alpha}_t(i)\cdot a_{ij}\cdot b_i(x_{t+1})\cdot\hat{\beta}_j(t+1) \tag{3.57c}$$

$$\gamma_t(i) = \frac{\alpha_t(i)\cdot\beta_t(i)}{P(X_1^T|\lambda)} \tag{3.58a}$$

$$= \frac{C_t\cdot\alpha_t(i)\cdot\beta_t(i)\cdot D_t}{c_t} \tag{3.58b}$$

$$= \frac{\hat{\alpha}_t(i)\cdot\hat{\beta}_t(i)}{c_t} \tag{3.58c}$$

O bien en la forma[2]

$$\gamma_t(i) = \sum_{j=1}^{N}\xi_t(i,j) \tag{3.59a}$$

$$= \sum_{j=1}^{N}\hat{\alpha}_t(j)\cdot a_{ij}\cdot b_j(x_{t+1})\cdot\hat{\beta}_{t+1}(j) \tag{3.59b}$$

Compresión logarítmica en el algoritmo de Viterbi

Para el algoritmo de Viterbi puede desarrollarse también un escalado similar al utilizado para el *Forward-Backward* [Peinado89], sin embargo, es más sencillo y menos costoso modificar el mismo para utilizar una representación logarítmica de las probabilidades en la forma

---

[2]Es sencilo comprobar (3.59) de la definición (3.31a)

Algoritmo de Viterbi modificado

1) *Inicialización:*

$$\hat{\delta}_t(i) = \hat{\pi}_i + \hat{b}_i(x_1) \qquad 1 \le i \le N \qquad (3.60a)$$

2) *Recursión:*

$$\hat{\delta}_t(i) = \max_{1 \le j \le N} \{\hat{\delta}_{t-1}(j) + \hat{a}_{ji}\} + \hat{b}_i(x_t) \qquad \begin{matrix} t=1,2,\cdots,T \\ 1 \le i \le N \end{matrix} \qquad (3.60b)$$

$$\psi_t(i) = \operatorname*{argmax}_{1 \le j \le N} \{\hat{\delta}_{t-1}(j) + \hat{a}_{ji}\} \qquad \begin{matrix} t=1,2,\cdots,T \\ 1 \le i \le N \end{matrix} \qquad (3.60c)$$

3) *Terminación:*

$$\log P^*(X_1^T|\lambda) = \max_{1 \le i \le N} \{\hat{\delta}_T(i)\} \qquad (3.60d)$$

$$y_T^* = \operatorname*{argmax}_{1 \le i \le N} \{\hat{\delta}_T(i)\} \qquad (3.60e)$$

4) *Recursión para obtener la secuencia de estados:*

$$y_t^* = \psi_{t+1}(y_{t+1}^*) \qquad t=T-1,T-2,\cdots,1 \qquad (3.60f)$$

Donde

$$\hat{\pi}_i = \log \pi_i \qquad 1 \le i \le N \qquad (3.61a)$$

$$\hat{a}_{ij} = \log a_{ij} \qquad \begin{matrix} 1 \le i \le N \\ 1 \le j \le N \end{matrix} \qquad (3.61b)$$

$$\hat{b}_i(x) = \log b_i(x) \qquad 1 \le i \le N \qquad (3.61c)$$

Esto es válido ya que la función logaritmo es estrictamente creciente, y en consecuencia

$$\log\left[\max_x \{f(x)\}\right] = \max_x \{\log f(x)\} \qquad (3.62)$$

3.5.2. MULTIPLES SECUENCIAS DE OBSERVACIONES

En muchas situaciones, en las que por la naturaleza de los procesos a modelar, éstos son de duración finita (ej. pronunciación de una palabra o fonema), no se dispone de una secuencia de observaciones de la suficiente duración como para poder estimar adecuadamente los parámetros del modelo, por lo que es necesario utilizar varias secuencias de observaciones en la estimación de dichos parámetros. Dado que las fórmulas de reestimación están desarrolladas para una secuencia simple de observaciones, es necesario modificarlas para poder tener en cuenta un conjunto de secuencias.

Como las fórmulas de reestimación (3.33) están expresadas es términos de frecuencias de ocurrencia de determinados sucesos, y suponiendo independencia estadística entre las diferentes secuencias de observaciones, se puede simplemente acumular los valores de las frecuencias de ocupación y de transición de estados para todas las secuencias del conjunto de entrenamiento, con lo que las fórmulas de reestimación quedan modificadas como sigue

$$\bar{\pi}_i = \sum_{l=1}^{L} \gamma_1^{(l)}(i) \qquad 1 \le i \le N \tag{3.63a}$$

$$\bar{a}_{ij} = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T-1} \xi_t^{(l)}(i,j)}{\sum_{l=1}^{L}\sum_{t=1}^{T-1} \gamma_t^{(l)}(i)} \qquad \begin{matrix} 1 \le i \le N \\ 1 \le j \le N \end{matrix} \tag{3.63b}$$

$$\bar{b}_i(k) = \frac{\sum_{l=1}^{L}\sum_{\substack{t=1 \\ x_t = v_k}}^{T} \gamma_t^{(l)}(i)}{\sum_{l=1}^{L}\sum_{t=1}^{T} \gamma_t^{(l)}(i)} \qquad \begin{matrix} 1 \le i \le N \\ 1 \le k \le M \end{matrix} \tag{3.63c}$$

donde el superíndice (l) indica que los valores han sido obtenidos para la secuencia de observaciones l, perteneciente a un conjunto de L secuencias

$$\Xi = \{X_1^{T_1}, X_1^{T_2}, \cdots, X_1^{T_L}\} \tag{3.64a}$$

$$X_1^{T_l} = x_1^{(l)}, x_2^{(l)}, \cdots x_{T_l}^{(l)} \tag{3.64b}$$

3.5.3. ENTRENAMIENTO INSUFICIENTE

El conjunto de secuencias de entrenamiento, necesariamente finito, es frecuentemente demasiado reducido para obtener estimaciones adecuadas de los parámetros de los modelos. Este problema es especialmente importante en la estimación de las probabilidades de producción de símbolos en los modelos discretos, y en la estimación de los elementos de las matrices de covarianza de las densidades de probabilidad de observaciones en el caso de los modelos contínuos.

Para evitar el problema del entrenamiento insuficiente de los modelos, sólo se puede aumentar el conjunto de secuencias de entrenamiento, o disminuir el número de parámetros a estimar en los modelos. Cuando ninguna de estas alternativas es posible, se puede minimizar este efecto interpolando un modelo $\bar{\lambda}$ en base al modelo estimado λ, y a otro $\tilde{\lambda}$, en el que las probabilidades están suavizadas, que se puede obtener de diferentes formas. El proceso de interpolación está controlado por un parámetro ε en la forma

$$\bar{\lambda} = \varepsilon \cdot \lambda + (1-\varepsilon) \cdot \tilde{\lambda} \quad ; \quad \varepsilon \in [0,1] \tag{3.65}$$

El parámetro de interpolación ε puede obtenerse mediante un método de prueba y error, o bien de forma automática mediante un proceso de reestimación *Forward-Backward*. Este método consiste en utilizar un diagrama de transiciones como el de la figura 3.7, en base a las probabilidades de transición entre un estado neutral $\tilde{s}$ y los dos modelos a interpolar λ y $\tilde{\lambda}$. Para obtener los modelos, se utiliza una partición binaria disjunta de la secuencia de entrenamiento, por ejemplo T_1 y T_2, de forma que la primera se utiliza para la estimación de los modelos λ y $\tilde{\lambda}$, y la segunda para la estimación de ε.

Una generalización de este método, denominado *Deleted interpolation* [Rabiner89a, Jelinek80], consiste en construir n particiones binarias disjuntas de la secuencia de entrenamiento para poder estimar ε con un número suficientemente grande de secuencias. Por ejemplo, se puede particionar un conjunto de 1000 secuencias de

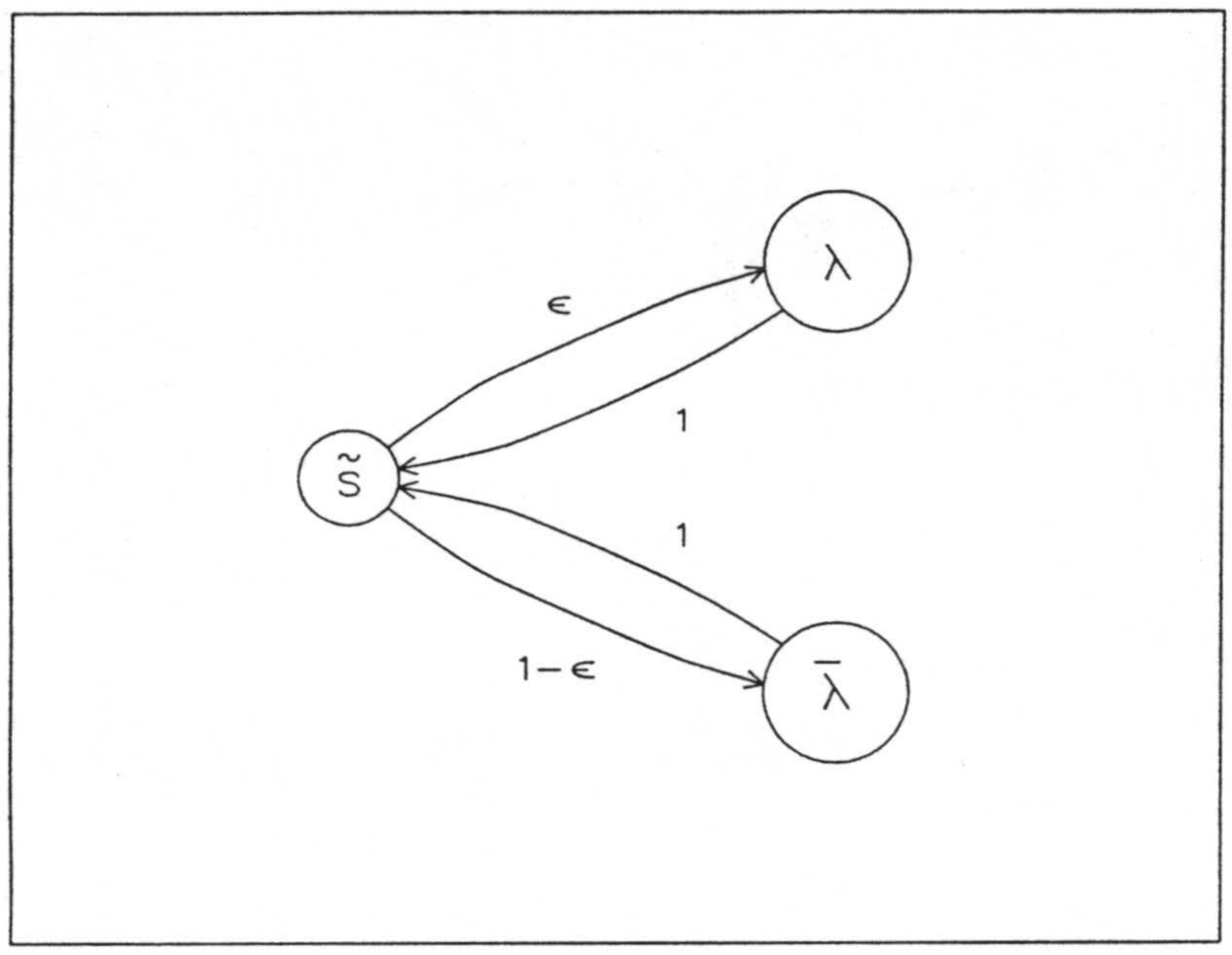

Figura 3.7. Diagrama de estados para el proceso de interpolación

entrenamiento de forma que una partición contenga 900 secuencias para estimar los modelos y 100 para estimar ε, de forma rotativa, obteniendo 10 particiones del conjunto de secuencias de entrenamiento, y utilizando efectivamente las 1000 secuencias para la estimación de ε.

Distribuciones uniformes

La forma más sencilla de obtener el modelo $\tilde{\lambda}$ a interpolar es suponer que sus distribuciones de probabilidad son uniformes. En este caso, el proceso es equivalente a imponer la restricción de que las probabilidades no sean en ningún caso inferiores a un cierto valor δ prefijado

$$b_i(k) \geq \delta \quad ; \quad \begin{matrix} 1 \leq i \leq N \\ 1 \leq k \leq M \end{matrix} \tag{3.66}$$

asignando el valor δ a todas aquellas probabilidades inferiores a este valor después de cada reestimación, y escalando el resto para preservar la condición de normalización. El resultado obtenido es equivalente a interpolar un modelo con distribuciones de probabilidad

uniforme con un valor $\varepsilon = (1-\delta)$.

Modelos reducidos

Otra posible elección para $\tilde{\lambda}$ es construir un modelo en el que determinados parámetros del modelo λ están compartidos [Jelinek80] (ej. dos o más distribuciones de probabilidad entre dos o más estados).

Suavizado coocurrente

Este método [Lee89] consiste en utilizar información sobre la probabilidad de que un símbolo sea reemplazado por otro durante el proceso de producción del modelo a fin de construir las distribuciones de probabilidad suavizadas para el modelo a interpolar, para lo que se define la matriz de coocurrencia de símbolos en la forma

$$c(i|j) = \frac{\sum_{m=1}^{N_m} \sum_{s=1}^{N_s(m)} P(v_i|s,\lambda_m)\cdot P(v_j|s,\lambda_m)\cdot P(s|\lambda_m)\cdot P(\lambda_m)}{\sum_{k=1}^{M} \sum_{m=1}^{N_m} \sum_{s=1}^{N_s(m)} P(v_k|s,\lambda_m)\cdot P(v_j|s,\lambda_m)\cdot P(s|\lambda_m)\cdot P(\lambda_m)} \tag{3.67}$$

que representa la probabilidad de que, dada la observación de un símbolo v_j, se observe el símbolo v_i en un contexto similar dado por el estado s y el modelo λ_m. Estos valores pueden calcularse a partir de los modelos estimados λ_m en la forma

$$c(i|j) = \frac{\sum_{m=1}^{N_m} \sum_{s=1}^{N_s(m)} b_s{}^{(m)}(i)\cdot b_s{}^{(m)}(j)\cdot P(s|\lambda_m)\cdot P(\lambda_m)}{\sum_{k=1}^{M} \sum_{m=1}^{N_m} \sum_{s=1}^{N_s(m)} b_s{}^{(m)}(k)\cdot b_s{}^{(m)}(j)\cdot P(s|\lambda_m)\cdot P(\lambda_m)} \tag{3.68}$$

donde el superíndice m recorre el conjunto de modelos diseñados, $P(s|\lambda_m)$ es la probabilidad del estado S en el modelo λ_m, y $P(\lambda_m)$ es la probabilidad incondicional del modelo λ_m.

Con estos valores, se pueden obtener las probabilidades del modelo $\tilde{\lambda}$ en la forma

$$\tilde{b}_s^{(m)} = \sum_{j=1}^{M} c(i|j) \cdot b_s^{(m)}(j) \quad ; \quad \begin{matrix} 1 \le m \le N_m \\ 1 \le s \le N \\ 1 \le i \le M \end{matrix} \tag{3.69}$$

CAPITULO 4

RECONOCIMIENTO DE VOZ CON MODELOS HMM

4.1. MODELADO HMM DE SEÑALES DE VOZ

La señal de voz es una señal acústica cuyas características espectrales varían contínuamente en el tiempo, de forma que no es posible realizar un análisis espectral estacionario. Sin embargo, dado que esta variación es relativamente lenta, es posible asumir que las características de la señal, en un intervalo suficientemente corto de tiempo (del orden de 20 milisegundos), no varían apreciablemente, de forma que es posible realizar un análisis espectral cuasi-estacionario sobre dichos segmentos de señal. La evolución temporal de las características espectrales se obtiene repitiendo el proceso de análisis sobre segmentos consecutivos de la señal. A la serie de espectros obtenidos para segmentos consecutivos de una señal que varía en el tiempo se le suele denominar *espectrograma de la señal.* En la figura 4.1 se muestran los espectrogramas de dos palabras castellanas. El eje de ordenadas representa el tiempo, y el de abcisas la frecuencias, mientras que el nivel de gris representa el logaritmo de la energía del espectro.

En la figura puede apreciarse que los dos espectrogramas presentan zonas homogéneas en las que las características espectrales no varían de forma significativa, intercaladas entre las cuales se observan zonas claramente transicionales.

Para modelar una señal de voz, es necesario modelar la evolución temporal de las características espectrales de la misma. Esto puede realizarse mediante un modelo oculto de Markov asociando los estados del modelo a las zonas estacionarias (y posiblemente también las transicionales) de la señal de forma que las probabilidades de producción de observaciones modelen la variabilidad estadística de las características espectrales de cada zona, mientras que las probabilidades de transición modelan su secuenciamiento y

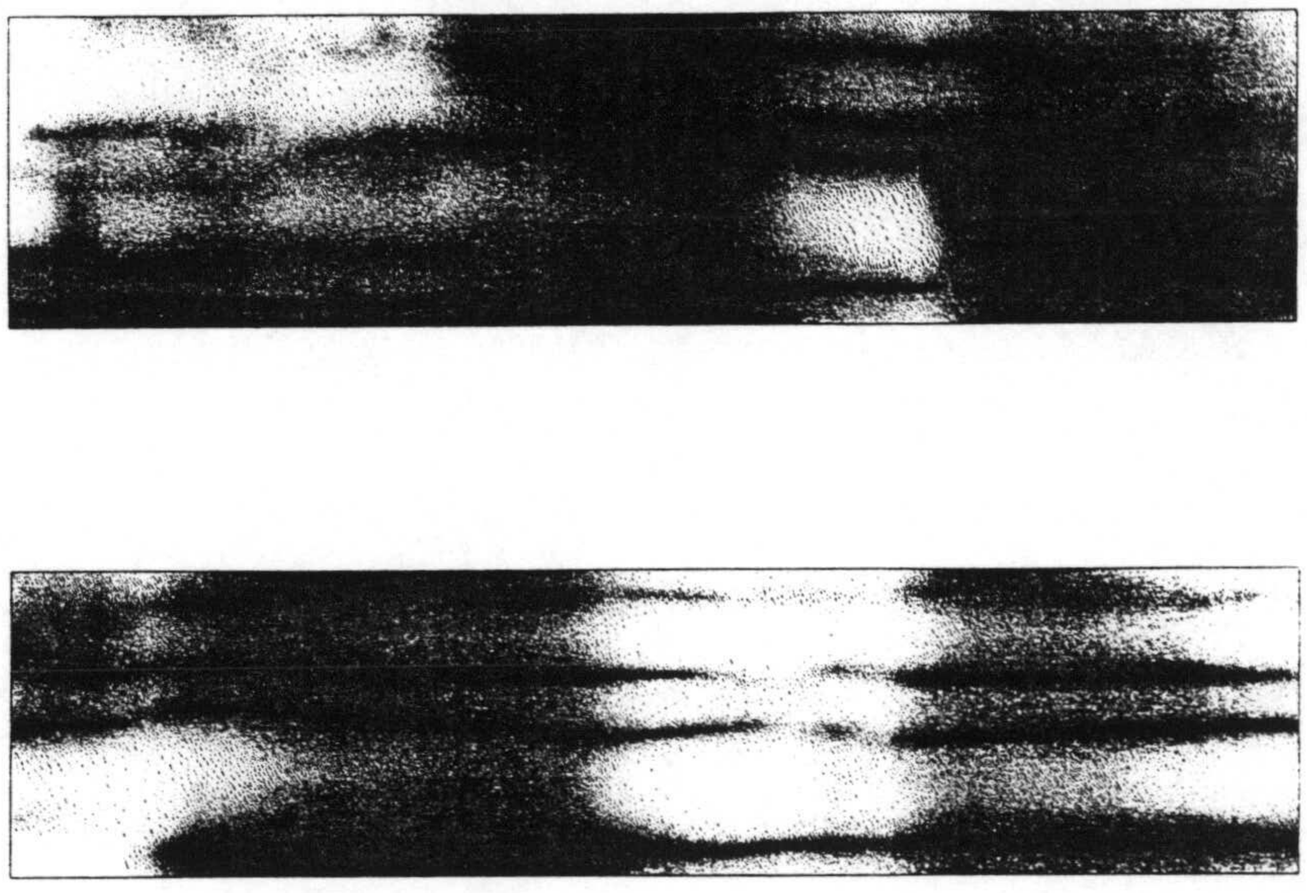

Figura 4.1. Espectrogramas de /MUÑECA/ (arriba) y /SIETE/ (abajo)

duración.

Por estas razones, la topología usualmente elegida para los modelos HMM es la denominada *izquierda-derecha*. Esta topología fue inicialmente propuesta por Bakis [Bakis76] y se muestra esquemáticamente en la figura 4.2. Este modelo consiste en una

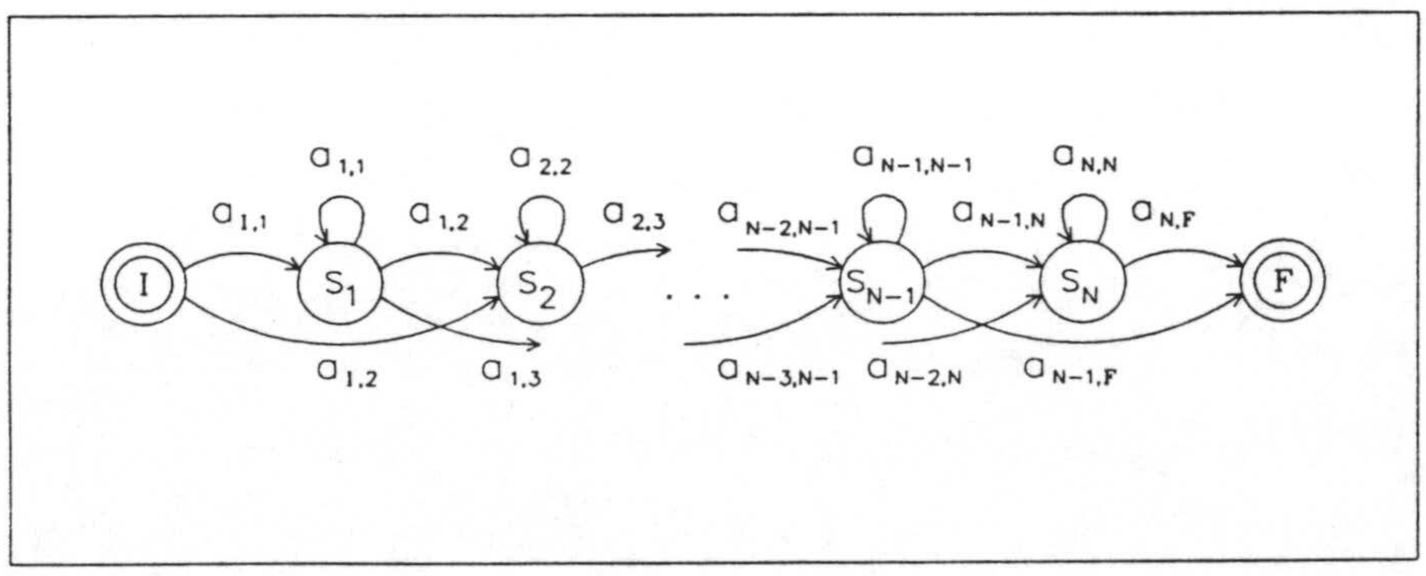

Figura 4.2. Modelo oculto de Markov *izquierda-derecha*

secuencia de N estados y dos estados especiales, I inicial y F final. Estos estados son especiales en el sentido de que no generan observaciones, a diferencia del resto de los estados del modelo. Además, sólo están permitidas transiciones hacia delante, es decir, que las probabilidades de transición son tales que

$$a_{ij} = 0 \qquad j<i \ , (j-i)>\Delta \tag{4.1}$$

donde Δ es un parámetro que controla el número de estados que pueden ser "saltados" por el modelo durante su evolución temporal.

Modelos con topologías como la descrita anteriormente se han aplicado para el modelado de unidades constitutivas del habla tales como palabras [Rabiner85a, Gupta87], fonemas [Lee87a, Chown87], y unidades relacionadas, como demisílabas [Mariño90].

4.2. RECONOCIMIENTO DE VOZ MEDIANTE MODELOS HMM

El reconocimiento de voz mediante modelos ocultos de Markov se basa en la identificación de unidades constitutivas del habla a través de la probabilidad de generación de las secuencias de observaciones de dichas unidades, obtenidas de un modelo HMM previamente construido para cada una de las unidades a reconocer (palabras, fonemas, etc.).

A la fase en la que se construyen los modelos se la denomina usualmente fase de entrenamiento, y se realiza en base a un conjunto de observaciones obtenidas para cada uno de los diferentes tipos de unidades a modelar. Usualmente se utiliza un algoritmo de estimación de máxima probabilidad como el Baum-Welch, pero en la bibliografía pueden encontrarse formulaciones sobre la reestimación de los parámetros de los modelos como la estimación de máxima información mútua MMI [Chow90], entrenamientos competitivos [Appelbaum89] o del estilo de los utilizados en redes neuronales [Young90], que formulan el problema de la estimación global de los parámetros de un conjunto de modelos HMM de forma que se maximice el potencial discriminativo de éstos.

También se han formulado otras aproximaciones al modelado HMM en las que la topología del modelo se determina a partir de las secuencias de entrenamiento, en lugar de seleccionarlo *a priori*, como en el algoritmo ECGI [Torró-Enguix90], o mediante técnicas de alineamiento temporal de las secuencias de entrenamiento [Falaschi90].

Una vez construidos los modelos en la fase de entrenamiento, y dada una secuencia de observaciones incógnita X_1^T, el proceso de reconocimiento necesita de la selección del modelo más probable dada dicha secuencia de observaciones, es decir que se asume que la secuencia corresponde al modelo λ_x para el que se verifica

$$\lambda_x = \max_{\lambda \in \Lambda} \{P(\lambda | X_1^T)\} \tag{4.2}$$

donde $\Lambda = \{\lambda_i\}_{i=1..L}$ es el conjunto de modelos correspondientes a las diferentes unidades a reconocer.

Estas probabilidades *a priori* pueden calcularse en base a las probabilidades *a posteriori* de generación de la secuencia por parte de los modelos según la regla de Bayes

$$P(\lambda_i | X_1^T) = \frac{P(X_1^T | \lambda_i) \cdot P(\lambda_i)}{\sum_{j=1}^{L} P(X_1^T | \lambda_j) \cdot P(\lambda_j)} \qquad 1 \le i \le L \tag{4.3}$$

donde $P(\lambda_i)$ es la probabilidad incondicional de ocurrencia del modelo λ_i.

Dado que el denominador de (4.2) es constante, basta con calcular las probabilidades *a posteriori* $P(X_1^T | \lambda_i)$ con un algoritmo como el descrito en la sección 3.3.1. Además, si suponemos equiprobables *a priori* todos los modelos, entonces el factor $P(\lambda_i)$ es también constante, y la regla de decisión se reduce a

$$\lambda_x = \max_{\lambda \in \Lambda} \{P(X_1^T | \lambda)\} \tag{4.4}$$

4.3. ANALISIS Y CARACTERIZACION DE LA SEÑAL DE VOZ

El primer paso para el modelado de la voz mediante modelos HMM es el de la extracción de características de la misma que se puedan utilizar como observaciones del proceso. Para ésto, usualmente se realiza un análisis espectral de la señal extrayendo parámetros que caractericen las propiedades espectrales de la misma. Como ya indicamos anteriormente, el análisis se basa en la caracterización del espectro de la señal en segmentos cortos consecutivos de la misma.

En el presente trabajo utilizaremos una técnica de análisis de predicción lineal LPC [Markel76a, Rabiner78] basada en un modelo de producción de voz como el de la figura 4.3 en el que la señal de voz se modela como una señal excitación $u(n)$ periódica (en el caso de sonidos sonoros) o con características de ruido blanco gausiano (en el caso de sonidos no sonoros), modificada por un filtro lineal todo polos de coeficientes constantes $H(z)$ de la forma

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 + \sum_{k=1}^{p} a_k \cdot z^{-k}} \tag{4.5}$$

donde $A(z)$ se denomina filtro de error de predicción o filtro inverso, $\{a_k\}_{k=1..p}$ son los coeficientes de predicción lineal o coeficientes LPC, y p es el orden de predicción del filtro. El resto de los parámetros del sistema son la frecuencia fundamental, que en el caso de excitación periódica representa el periodo de dicha excitación; la decisión sonoro/no-sonoro, que determina el tipo de excitación; y por último, la ganancia del sistema, que determina básicamente la energía de la señal obtenida $s(n)$.

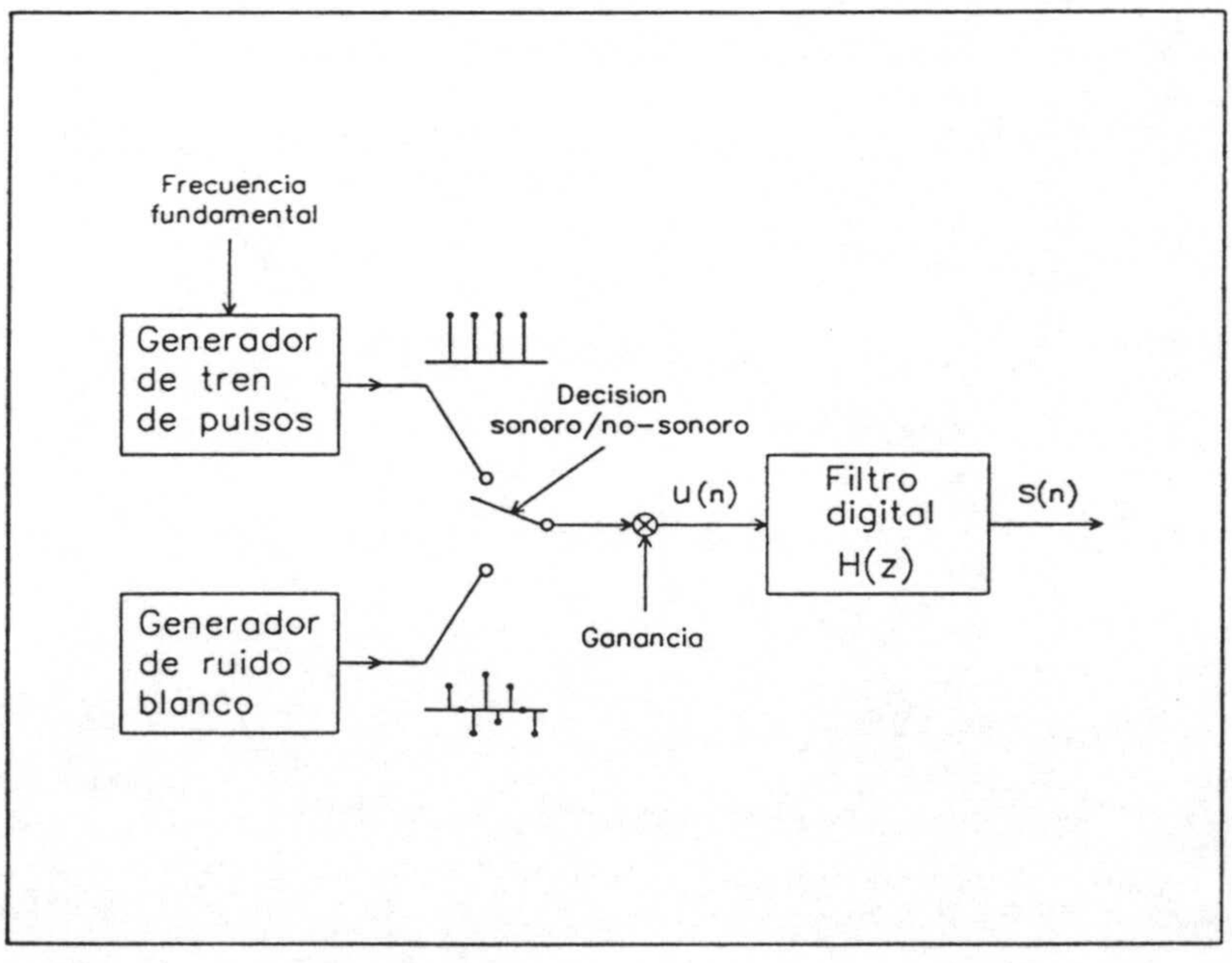

Figura 4.3. Modelo de producción de voz

Durante el análisis se supone que, para cada segmento consecutivo de señal considerado, los parámetros del modelo son constantes, y sólo varían entre diferentes segmentos correspondientes a los diferentes instantes de tiempo considerados, de forma que se realiza un análisis estacionario para cada segmento de señal considerado.

De este análisis, se puede extraer la envolvente del espectro de la señal, que caracteriza adecuadamente al espectro de la misma, y viene dada por la función de transferencia $H(z)$ del filtro lineal, dado que el espectro de la excitación $u(n)$ es plano; eliminando la estructura fina del mismo, correspondiente a dicha excitación. Esta separación de las contribuciones espectrales es adecuada sobre todo cuando el propósito es la comparación espectral de dos segmentos de señal, que se realiza adecuadamente en base a sus envolventes espectrales.

En base a los coeficientes LPC extraidos de esta forma, se han formulado medidas de distorsión espectral como la de *Itakura-Saito* o la *Likelihood ratio* [Buzo80a, Soong88], utilizadas en el proceso de cuantización vectorial (ver más adelante) necesario para la implementación de los HMM discretos [Rabiner83a, Peinado89], y densidades de probabilidad para observaciones de modelos HMM contínuos [Juang85]. Sin embargo, recientes trabajos han mostrado que los coeficientes del cepstrum [Oppenheim75] de la señal son un conjunto de parámetros más adecuado para la representación de la envolvente espectral de la misma en aplicaciones de reconocimiento de voz [Rabiner85b, Furui86] y verificación e identificación de locutores [Furui81a, Atal74]

Los coeficientes cepstrum que contribuyen a la envolvente del espectro de la señal pueden obtenerse a través de un análisis homomórfico de la misma, del que se puede obtener una relación recursiva que relaciona los coeficientes cepstrum de la respuesta en frecuencia de un sistema causal estable con los coeficientes de su respuesta al impulso [Oppenheim75] de la forma

$$c(n) = \begin{cases} 0 & ; n<0 \\ \log h(0) & ; n=0 \\ \dfrac{h(n)}{h(0)} - \displaystyle\sum_{k=0}^{n-1} \frac{k}{n} c(k) \frac{h(n-k)}{h(0)} & ; n>0 \end{cases} \tag{4.6}$$

donde $h(n)$ son las muestras de la respuesta al impulso del filtro lineal que caracteriza al espectro de la señal.

La respuesta al impulso del filtro inverso $A(z)$ es finita y sus valores están definidos en función de los coeficientes de predicción en la forma

$$h(n) = \begin{cases} 0 & ; n<0 \\ 1 & ; n=0 \\ a(n) & ; 1 \leq n \leq p \\ 0 & ; n>p \end{cases} \tag{4.7}$$

Además, los espectros logarítmicos de $H(z)$ y $A(z)$ son iguales salvo un cambio de signo, y como la dependencia del espectro logarítmico con los coeficientes cepstrum es lineal, se produce un cambio de signo en dichos coeficientes, de forma que se puede escribir la relación siguiente para la obtención de los cepstrum correspondientes al espectro LPC de la señal en la forma

$$c(n) = \begin{cases} 0 & ; n \leq 0 \\ -a(1) & ; n=1 \\ -a(n) - \displaystyle\sum_{k=1}^{n-1} \frac{k}{n} c(k) a(n-k) & ; n>1 \end{cases} \tag{4.8}$$

La envolvente del espectro logarímico de la señal puede calcularse entonces en términos de los coeficientes cepstrum en la forma

$$F(e^{jw}) = \log H(e^{jw}) = \sum_{n=-\infty}^{+\infty} c(n) \cdot e^{-jwn} \tag{4.9}$$

4.4. DISCRETIZACION DEL ESPACIO DE CARACTERISTICAS

Una vez definida la topología del modelo, y los parámetros que determinan el vector de observaciones del proceso, queda totalmente especificado el problema del modelado HMM contínuo, salvo la elección de la forma de las funciones densidad de probabilidad a utilizar en la mezcla que modele las probabilidades de las observaciones en los estados del modelo.

Sin embargo, para el modelado HMM discreto, es necesario que el conjunto de observaciones del proceso sea discreto, de forma que es necesario discretizar el espacio de vectores observables. Es decir, es necesario establecer una partición de dicho espacio en un conjunto finito de clases de forma que quede unívocamente definida la clase a la que pertenece cada vector de observaciones, con lo que se podrá utilizar dicho conjunto de clases como observaciones discretas del proceso. El problema puede enunciarse de la siguiente forma

> *Dado un conjunto de vectores* $\{x_i\}_{i=1..n}$, *cómo seleccionar un conjunto de clases* $\{v_i\}_{i=1..M}$ *que represente adecuadamente dicho conjunto de vectores.*

La solución a este problema, cuando el conjunto de vectores $\{x_i\}_{i=1..n}$ es una representación estadísticamente significativa del conjunto de posibles valores de los vectores de observación, nos llevará a un conjunto de clases que representa adecuadamente el espacio de características.

Cuando el conjunto de clases está representado por un subconjunto discreto de vectores del espacio, y el problema de clasificación consiste en la sustitución de cada vector del espacio por el representante de cada clase, estamos ante un problema de cuantización vectorial bien conocido en el contexto de codificación de señales, denominado usualmente VQ (de las iniciales de la denominación inglesa Vector Quantization) [Gray84a, Makhoul85a, Buzo80], y cuyos resultados han sido aplicados tradicionalmente en el reconocimiento de voz con modelos HMM discretos [Rabiner83].

El problema antes expuesto puede formularse como un problema de agrupamiento (clustering en la bibliografía inglesa) [Duda73b]. Los procedimientos de *clustering* generan una descripción de un conjunto de datos en base a *clusters* o agrupaciones de datos con fuertes similitudes entre si.

Una definición formal del problema necesita de la especificación de un criterio de agrupamiento, de forma que el procedimiento de búsqueda de clases extremice dicho criterio, y que a su vez se formula a través de una medida de similitud entre vectores del espacio.

4.4.1. MEDIDA DE SIMILITUD ESPECTRAL

Dado que los vectores observables representan características espectrales de la señal, una medida de similitud entre éstos debe formularse en base a similitudes espectrales entre los segmentos de señal a los que corresponden. En la literatura se han formulado diversas medidas de similitud espectral usualmente denominadas medidas de *distorsión espectral*, como las distancias de Itakura-Saito o razón de semejanza [Gray80], pero una de las que mejores resultados ofrece en cuanto al reconocimiento de voz e identificación y verificación de locutores es la diferencia cuadrática media entre los espectros logarítmicos de los segmentos de señal de voz a comparar.

Supongamos que $H_r(e^{jw})$ y $H_s(e^{jw})$ representan las envolventes espectrales de dos segmentos de señal, entonces la diferencia cuadrática media entre los correspondientes espectros logarítmicos $F_r(e^{jw})$ y $F_s(e^{jw})$ puede calcularse en la forma

$$d(F_r, F_s) = \int_{-\pi}^{+\pi} \left| \log H_r(e^{jw}) - \log H_s(e^{jw}) \right|^2 \frac{dw}{2\pi} \qquad (4.10)$$

Esta distancia puede expresarse en términos de los coeficientes cepstrum correspondientes utilizando (4.8) en la forma

$$d(F_r, F_s) = \int_{-\pi}^{+\pi} \left| \sum_{n=-\infty}^{+\infty} [c_r(n) - c_s(n)] e^{-jwn} \right|^2 \frac{dw}{2\pi} \tag{4.11a}$$

$$= \int_{-\pi}^{+\pi} \sum_{n=-\infty}^{+\infty} \sum_{m=-\infty}^{+\infty} [c_r(n) - c_s(n)] \cdot [c_r(m) - c_s(m)] \cdot e^{-jw(n+m)} \frac{dw}{2\pi} \tag{4.11b}$$

$$= \sum_{n=-\infty}^{+\infty} [c_r(n) - c_s(n)]^2 \tag{4.11c}$$

y puede aproximarse mediante una suma finita de coeficientes cepstrum, de forma que dicha distancia espectral se convierte en una distancia euclídea en el espacio de vectores de observación formados por los coeficientes cepstrum de las envolventes espectrales. Así, definiremos la distancia entre vectores de coeficientes cepstrum en la forma

$$d_{CEP}(x_r, x_s) = \sum_{n=1}^{p} [c_r(n) - c_s(n)]^2 = \| x_r - x_s \|^2 \tag{4.12}$$

donde x_r y x_s representan los dos vectores de parámetros a comparar y p es la dimensión del vector (número de coeficientes cepstrum considerados).

La elección de esta distancia justifica la elección de los coeficintes cepstrum como parámetros observables. A diferencia de otras medidas de distorsión espectral como las de Itakura-Saito o razón de semejanza, ésta se convierte en una distancia euclídea en el espacio de características, lo que dota de interpretación geométrica al problema de agrupamiento en clases de dichos vectores y simplifica la formulación del mismo. Además, al estar formulada en base a los logaritmos de las envolventes espectrales, aproxima la escala de percepción del oido humano.

Por último, como veremos más adelante, esta formulación permite realizar de forma sencilla transformaciones sobre los logaritmos de los espectros tales como la ponderación espectral o la transformación de la escala de frecuencias, de forma que dichas transformaciones se reflejan de forma sencilla en la función distancia.

4.4.2. CRITERIO DE AGRUPAMIENTO

Una vez definida una medida de similitud en base a una distancia entre vectores del espacio de características, es sencillo definir un criterio de agrupamiento para la elección de las clases. Existen diferentes formas de definir dicho criterio [Duda73b] pero la más sencilla y comunmente utilizada es la de minimizar la distancia media entre el conjunto de vectores $\{x_i\}_{i=1..n}$ y un conjunto de clases $\{v_i\}_{i=1..M}$, en base a la suma de las distancias entre el conjunto de vectores y el conjunto de representantes de las clases $\{m_i\}_{i=1..M}$ en la forma

$$D = \frac{1}{n} \sum_{i=1}^{M} \sum_{x \in v_i} d(x, m_i) \tag{4.13}$$

En el caso concreto que nos ocupa, en el que la distancia es euclídea, los representantes de las clases que minimizan dicha distancia media son simplemente las medias aritméticas de los vectores que pertenecen a cada clase.

$$m_i = \frac{1}{n_i} \sum_{x \in v_i} x \qquad 1 \le i \le M \tag{4.14}$$

donde n_i es el número de vectores pertenecientes a la clase v_i.

El criterio de agrupamiento de vectores se formula simplemente en base a la minimización de la distancia media (4.12), y por lo tanto el criterio de asignación de clases a vectores es símplemente el de seleccionar la clase cuyo representante dista menos del vector considerado.

En general, cuando la medida de distorsión no es una distancia euclídea, los representantes de las clases no coinciden con las medias, y se denominan entonces *centroides* de las clases, y su cálculo depende de la forma de la medida de distorsión seleccionada.

El criterio de agrupamiento antes expuesto tiene la sencilla interpretación de la selección del conjunto de clases que minimiza el "error" cometido al sustituir cada vector x_i por el representante de la clase v_j de menor distorsión en base a la medida de dicha

distorsión. El valor de la distorsión media depende, evidentemente, de la forma en que se elija la partición en clases del conjunto de vectores y, consecuentemente, la partición óptima será aquella que minimice el valor de la distorsión media. Como ya indicamos antes, este criterio define el problema de cuantización vectorial, en el que al conjunto de representantes de las clases se le denomina *diccionario*, y al proceso de sustitución de los vectores por los representantes de las clases más similares se le denomina *cuantización vectorial.*

La obtención del diccionario permite asignar una clase a cada vector del espacio, y utilizar los centros del diccionario como símbolos de un alfabeto discreto, formado por diferentes tipos de envolventes espectrales. Aunque esta técnica de etiquetado (asignación de símbolos a segmentos), basada en cuantización vectorial, es la más comunmente utilizada en modelado HMM discreto de señales de voz, no es la única, y en la bibliografía pueden encontrarse otras técnicas relacionadas como la obtención de unidades microfonéticas en base a un algoritmo K-medias de dos pasos [Andreu90] o los mapas de Kohonen [Kohonen84].

4.4.3. CONSTRUCCION DEL DICCIONARIO

La construcción del diccionario necesita de la selección de un conjunto de vectores estadísticamente representativos del conjunto de posibles valores de los vectores del espacio de características (que en adelante denominaremos conjunto de entrenamiento), y de la partición de éste en un conjunto de clases que minimice la distorsión media D. Desafortunadamente, no existe solución analítica conocida a este problema más que la prueba exhaustiva de todas las posibles particiones, que conllevaría un cálculo proporcional al número de posibles particiones $M^n / M!$. Sin embargo, existen métodos iterativos que permiten determinar una partición subóptima, en el sentido de que obtienen un mínimo local para la distorsión media. Estos métodos se basan en la partición en clases del conjunto de entrenamiento y la reestimación de los representantes de las clases de forma iterativa hasta conseguir una configuración en la que dichas clases no varían de forma significativa. Algoritmos de este tipo son el *Isodata* y el algoritmo *K-medias* [Tou74], del que puede verse un diagrama de flujo en la figura 4.4.

Este algoritmo consiste en partir de una determinada partición en clases del conjunto de vectores de entrenamiento e iterar un proceso en el que los vectores son asignados a las

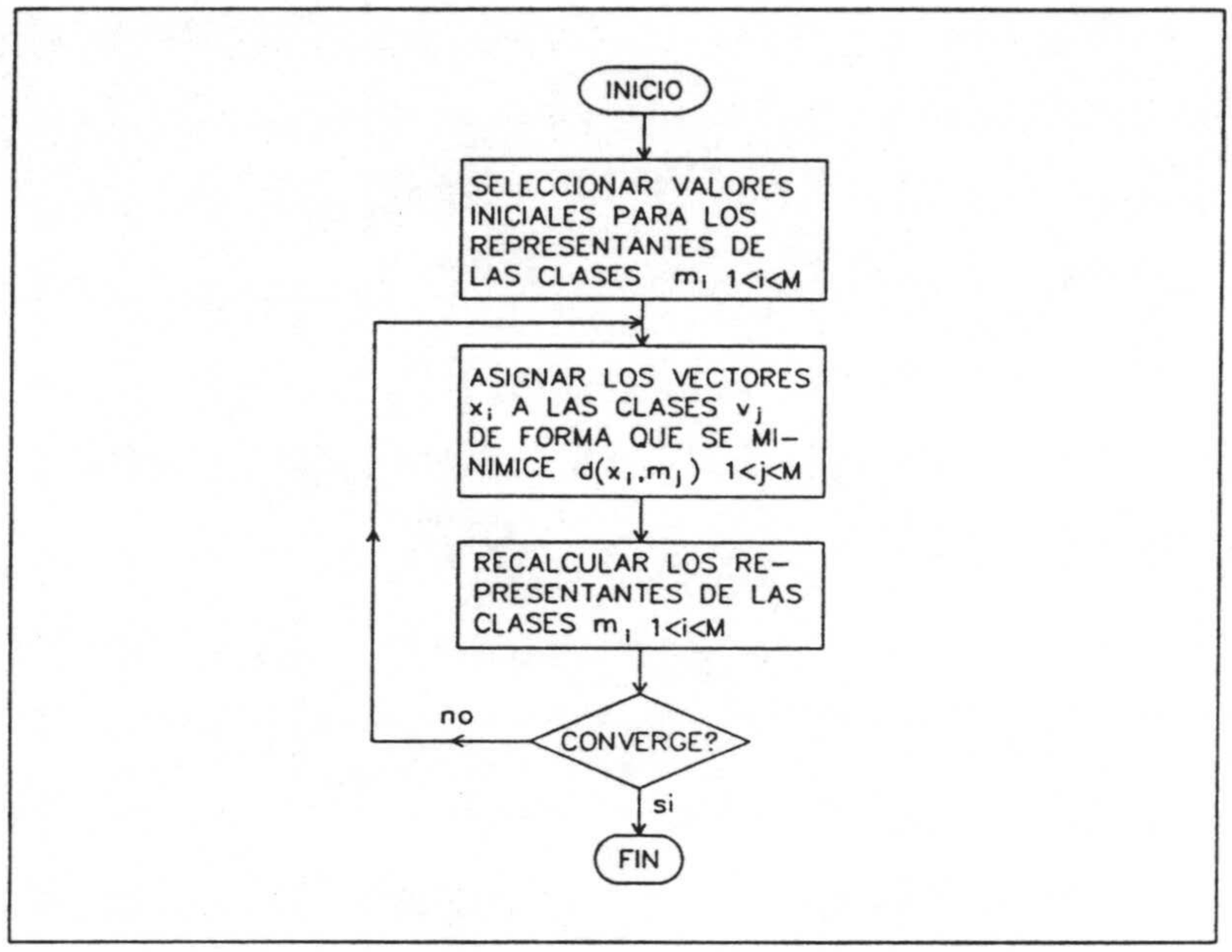

Figura 4.4. Algoritmo K-medias

clases, y una vez hecho ésto, recalcular los representantes de las clases, hasta conseguir un conjunto de clases que no varía entre una y otra iteración.

Desafortunadamente, el conjunto de representantes de clases obtenidos depende de la partición inicial considerada, y no existe forma conocida de determinar qué partición inicial lleva al algoritmo a la determinación del mínimo absoluto de la distorsión media. Para solucionar el problema de la elección de la partición inicial, se han propuesto algoritmos jerárquicos [Duda73b] que permiten seleccionar una partición incial de forma que el valor final obtenido para la distorsión media es cercano al mínimo absoluto. Tales algoritmos se basan en la obtención de una partición tal que, por el número de clases que la forman, es trivial, en el sentido de que se puede obtener la partición óptima para ese determinado número de clases, y a partir de ésta, generar nuevas particiones añadiendo o eliminando particiones con un criterio que minimice localmente la distorsión media en cada paso. Básicamente existen dos aproximaciones a dicha obtención jerárquica de las particiones que describimos a continuación.

Procedimientos aglomerativos

En este tipo de procedimientos, se parte de una partición tal que el número de clases coincide con el de vectores en el conjunto de entrenamiento. Evidentemente esta es una partición trivial dado que es única, y por lo tanto la óptima. A partir de ésta, se generan particiones con un número menor de clases agrupando cada vez las dos clases más similares de la partición previa, en base a un criterio de similitud de clases (p.e. la distancia entre los representantes de las clases), hasta llegar a una partición con el número deseado de clases.

El punto esencial en este método es que la unión de particiones está determinada por un criterio que minimiza la distorsión media de la nueva partición en base a la anterior. En cualquier caso es una minimización local que no conlleva necesariamente asociado el hecho de que la partición final obtenida corresponda con el mínimo obsoluto de la distorsión media.

Procedimientos divisivos

Los métodos anteriormente expuestos presentan el inconveniente del gran número de iteraciones necesarias para la obtención de una partición final con el número deseado de clases, cuando éste es muy inferior al número de vectores en el conjunto de entrenamiento.

Una alternativa a este tipo de métodos es la formada por los métodos divisivos. Como en el caso anterior, el punto inicial es una partición trivial formada, en este caso, por una única clase que agrupa a todos los vectores del conjunto de entrenamiento. Al igual que en el caso anterior, es óptima por ser única. A partir de esta configuración inicial, y en cada paso del algoritmo, se divide la clase con mayor distorsión media entre los vectores de la clase y su representante, de forma que también en este caso se minimiza localmente la distorsión media de la nueva partición. El procedimiento se itera hasta obtener el número deseado de clases.

En el presente trabajo utilizaremos un método de construcción jerárquica de diccionarios similar al propuesto para el algoritmo LGB [Linde80], basado en la división

sucesiva de todas las clases obtenidas en cada paso. En cada paso del algoritmo, y tras la división de los centros del diccionario, se aplica un algoritmo K-medias para obtener iterativamente los valores de los representantes de las clases que componen la partición, antes de proceder a una nueva división. En la figura 4.5 se muestra el diagrama de flujo del algoritmo utilizado en el presente trabajo.

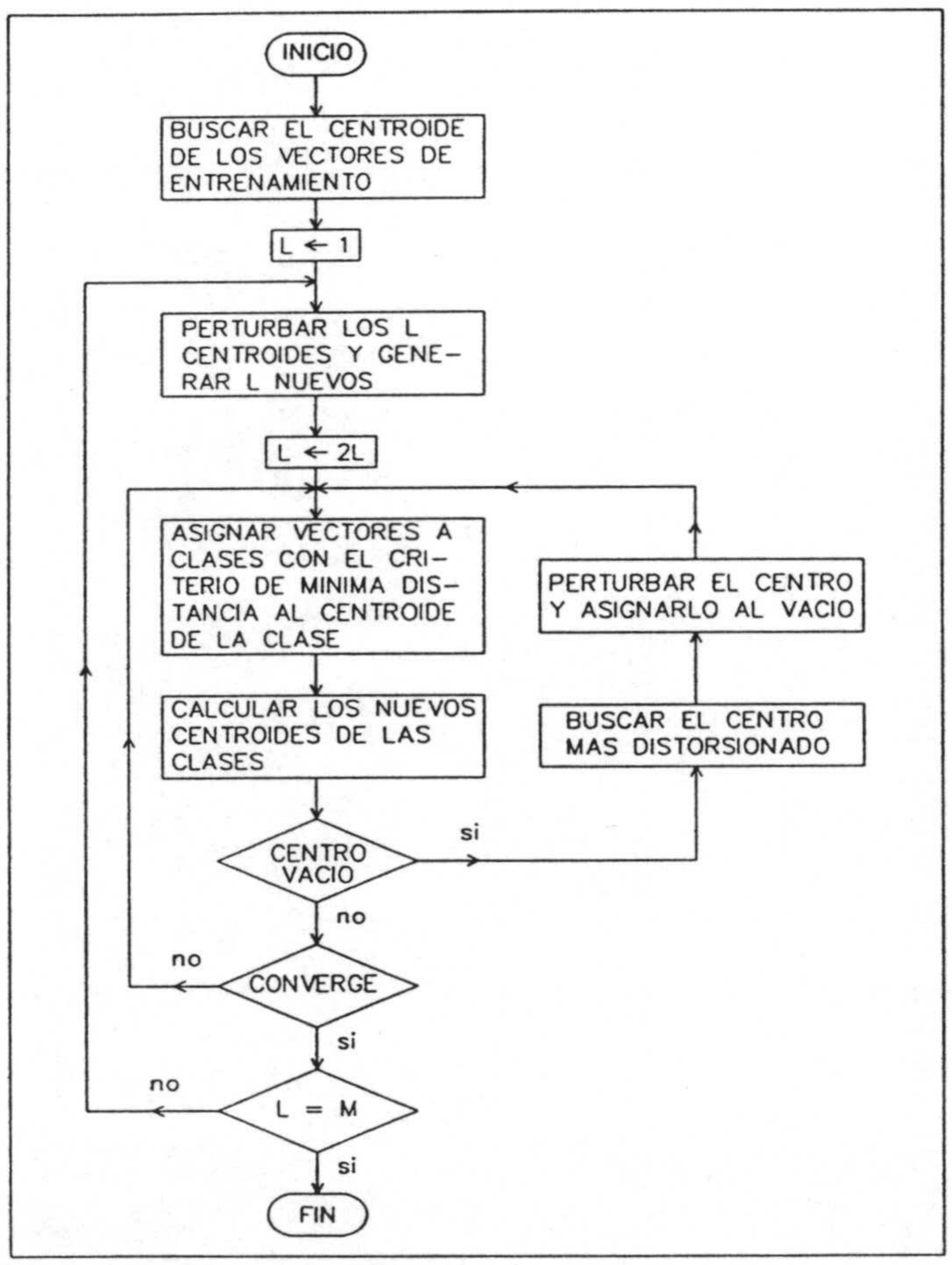

Figura 4.5. Algoritmo jerárquico de construcción de diccionarios

División de clases

La división de las clases se realiza en base a un método en el que, dado el centro de una clase $m_i^{(l)}$, es perturbado [Linde80], generando dos centros $m_i^{(l+1)}$ y $m_{l+i}^{(l+1)}$. En nuestro caso, el centro perturbado se obtiene multiplicando el centro original por un factor (1+μ), y obteniendo dos centros, el anterior, y el perturbado en la forma

$$\begin{aligned} m_i^{(l+1)} &= m_i^{(l)} \\ m_{l+i}^{(l+1)} &= (1+\mu)\cdot m_i^{(l)} \qquad 1 \le i \le l \end{aligned} \tag{4.15}$$

donde (l) denota el índice de la recursión para la obtención de las sucesivas particiones.

El valor de μ se elige pequeño de forma que se asegure que los vectores pertenecientes a la clase $v_i^{(l)}$ se distribuyan efectivamente entre las dos nuevas clases $v_i^{(l+1)}$ y $v_{l+i}^{(l+1)}$ cuando se obtengan iterativamente los representantes de éstas mediante el algoritmo K-medias antes citado.

Clases vacías

Es posible que, durante las iteraciones K-medias posteriores a un paso de división del algoritmo, una o más clases queden vacías (no tengan ningún vector asignado). Para solucionar este problema, el algoritmo busca la clase más distorsionada (con mayor distorsión media entre los vectores de la clase y su representante), y perturba su centro creando uno nuevo que es asignado a la clase vacía; repitiendo el proceso cada vez que es detectada una nueva clase vacía.

Criterio de convergencia

Para la terminación del algoritmo iterativo K-medias, se debe fijar un criterio con el que decidir cuando se ha alcanzado la configuración final. Este puede basarse en las variaciones de las configuraciones de las clases; bien en los elementos que pertenecen a cada una de ellas, determinando cuando no hay trasvase de vectores de unas clases a otras, o determinando cuando no se produce variación en los valores de los representantes de cada clase.

Una forma alternativa, es la de iterar hasta que no se produzca variación significativa en el valor de la distorsión media que determina el criterio de construcción de agrupamientos. Esto se consigue fijando un valor mínimo que determine qué variaciones en dicha distorsión media se considerarán significativas. De esta forma el proceso se itera hasta que la variación relativa en la distorsión media obtenida en dos iteraciones consecutivas es inferior a un cierto valor predeterminado, denominado usualmente parámetro de convergencia.

4.4.4. IMPLEMENTACION DEL ALGORITMO

En la implementación del algoritmo de construcción del diccionario que utlizaremos en el presente trabajo, se utiliza el procedimiento jerárquico divisivo mostrado en la figura 4.5 para obtener una configuración inicial, que posteriormente se refina mediante varias iteraciones del algoritmo K-medias antes expuesto, dando lugar a la configuración de centros definitiva del diccionario.

Para obtener rápidamente dicha configuración inicial, en cada paso de división se genera un nuevo centro por cada uno de los antiguos, doblando en cada paso el número de centros del diccionario. Esto obliga a que el número final de clases sea una potencia de dos. En contrapartida, el número de iteraciones de división necesarias es $\log_2 M$ frente a las M requeridas si en cada paso se genera unicamente un nuevo centro dividiendo únicamente el más distorsionado en cada iteración de división del algoritmo.

El valor de μ elegido es del orden de 0.001. En pruebas previas sobre experimentos de reconocimiento con HMM discretos, se comprobó que la tasa de error del sistema es relativamente insensible al valor concreto de este parámetro cuando su valor se encuentra

en este orden de magnitud.

En cuanto al parámetro de convergencia V_r, éste se fijó a 0.01 para las iteraciones K-medias realizadas durante el proceso de división para la obtención de la configuración inicial, y de 0.001 para las iteraciones K-medias realizadas con la configuración inicial obtenida mediante el proceso de iniciación jerárquica, para la obtención de los valores definitivos de los centros del diccionario. Esta elección acelera el proceso de determinación de la configuración inicial y, de nuevo, experimentos previos mostraron que la tasa de error es similar a la obtenida cuando se fijó 0.001 como parámetro de convergencia también para las iteraciones K-medias durante el proceso de iniciación.

Una vez construido el diccionario, los vectores de parámetros son reemplazados por símbolos de forma que cada vector se sustituye por el índice de la clase con el representante más cercano, de forma que se obtienen secuencias de símbolos discretos que representan las diferentes clases en las que se ha subdividido el espacio de vectores. De esta forma, el conjunto de clases obtenido pasa a ser el conjunto discreto de símbolos al que pertenecen las observaciones que se utilizarán en el modelado HMM discreto. Un modelo HMM discreto obtenido de esta forma tiene como producciones los diferentes tipos de envolventes espectrales que componen el diccionario, de forma que dicho HMM modela los diferentes tipos de espectros (clases o centros del diccionario) producidos durante la pronunciación de la unidad modelada.

4.5. MODIFICACIONES A LAS FORMULAS DE REESTIMACION

Las fórmulas de reestimación (3.63), cuando son aplicadas a modelos ocultos de Markov con topología *izquierda-derecha*, conllevan implícita la condición de que la probabilidad de transición entre el último estado del modelo s_N y éste mismo $a_{NN} = 1$ es la unidad, dado que no se tienen en cuenta las transiciones producidas entre este estado y el estado terminal F.

De otro lado, es sencillo comprobar que la distribución de probabilidad de las duraciones de los estados $P_i(d)$, que representa la probabilidad de que el modelo permenezca en el estado s_i durante d periodos consecutivos de tiempo, sigue una ley exponencial de la forma

$$P_i(d) = (a_{ii})^{d-1} \cdot (1-a_{ii}) \qquad 1 \leq i \leq N \tag{4.16}$$

y por lo tanto, la duración esperada $\bar{d}_i$ de cada estado s_i puede calcularse en función de las probabilidades de transición a_{ii} en la forma

$$\bar{d}_i = \sum_{i=1}^{\infty} d \cdot P_i(d) = \frac{1}{1-a_{ii}} \qquad 1 \leq i \leq N \tag{4.17}$$

Este resultado, junto con la condición $a_{NN} = 1$, nos lleva a la conclusión de que la duración esperada del último estado del modelo es infinita, lo que no es coherente con el hecho de que las secuencias de observaciones son finitas. Para corregir esta incoherencia, es necesario tener en cuenta las transiciones entre los estados del modelo y el estado terminal F. De esta forma asumiremos que siempre que, para una determinada secuencia de observaciones el estado final es el s_i, y éste puede realizar una transición al estado F, ésta transición ocurre (aunque no sea observada).

Para incorporar esta suposición a las fórmulas de reestimación de las probabilidades de transición, tendremos en cuenta la probabilidad de que un cierto estado s_i sea el último de una secuencia de estados para una secuencia de observaciones dada X_1^T, cuyo valor es simplemente

$$P(y_T = s_i \,|X_1^T, \lambda) = \gamma_T(i) \qquad 1 \leq i \leq N \tag{4.18}$$

con esta definición, podemos reformular la ecuación (3.63b) en la forma

$$\bar{a}_{ij} = \frac{\sum_{l=1}^{L} \sum_{t=1}^{T-1} \xi_t{}^{(l)}(i,j)}{\sum_{l=1}^{L} \left[\sum_{t=1}^{T} \gamma_t{}^{(l)}(i) - \gamma_T{}^{(l)}(i) \right]} \qquad \begin{array}{l} 1 \leq i \leq (N-\Delta) \\ 1 \leq j \leq N \end{array} \tag{4.19a}$$

$$\bar{a}_{Nj} = \frac{\sum_{l=1}^{L} \sum_{t=1}^{T-1} \xi_t{}^{(l)}(N,j)}{\sum_{l=1}^{L} \sum_{t=1}^{T} \gamma_t{}^{(l)}(i)} \qquad \begin{array}{l} (N-\Delta) < i \leq N \\ 1 \leq j \leq N \end{array} \tag{4.19b}$$

en las que, al sustituir el límite superior $T-1$ de la suma de $\gamma_t{}^{(l)}(i)$ por T, se asume que siempre que un estado es visitado, se produce una transición hacia otro estado del modelo, incluso para la última visita al estado final s_N del mismo. El valor $\gamma_T{}^{(l)}(i)$ se introduce para eliminar aquellas transiciones desde estados del modelo para los que no está permitida dicha transición al estado terminal F, considerando en estas situaciones que la secuencia de observaciones está incompleta, y teniendo en cuenta únicamente las transiciones observadas.

Es sencillo comprobar que, con estas definiciones, se preserva la condición de normalización de las probabilidades de transición de estados, si extendemos éstas a transiciones con el estado terminal F en la forma

$$\sum_{j=1}^{N} a_{ij} + a_{iF} = 1 \qquad 1 \leq i \leq N \tag{4.20}$$

y además se estiman las probabilidades de transición a_{NN} del último estado del modelo de forma coherente con la duración estimada de dicho estado.

Pruebas previas sobre un sistema standard de reconocimiento basado en HMM discretos han mostrado que el error de reconocimiento del sistema disminuye en torno a un 1% (sobre errores en torno al 9-10%) cuando se utilizan las fórmulas de reestimación (4.18) en sustitución de la presentada en (3.63b), para las probabilidades de transición de estados.

4.6. ESTIMACION INICIAL DE LOS PARAMETROS DEL MODELO HMM

Las fórmulas de reestimación para la obtención iterativa de los valores de los parámetros del modelo HMM sólo garantizan la obtención de un máximo local de la función probabilidad de generación del modelo. Por lo tanto, al igual que ocurre con el algoritmo K-medias, y en general con la mayoría los algoritmos iterativos, el resultado final obtenido depende de la elección inicial realizada para los parámetros a estimar.

No existe un método que permita seleccionar la configuración inicial para los parámetros de forma que se garantice la obtención de un máximo global de la función probabilidad, sin embargo la experiencia demuestra que una elección aleatoria o uniforme

para los valores de las probabilidades iniciales de estados Π y para las probabilidades de transición A resulta adecuada, sin embargo, ésto no es cierto para las probabilidades de producción de observaciones B.

En la bibliografía se pueden encontrar diversas formas de obtener estimaciones iniciales adecuadas para éstos parámetros, y que se basan en obtener una segmentación inicial de las secuencias de entrenamiento, y extraer de ésta los valores iniciales de todos los parámetros del modelo [Peinado91]. Para la obtención de esta segmentación se han utilizado métodos de segmentación manual, o lineal, en la que cada secuencia de entrenamiento es segmentada de forma que todos los estados tengan la misma duración. También se han propuesto otros métodos jerárquicos de segmentación basados en medidas de entropía de las distribuciones de probabilidad del modelo [Peinado91b], pero están todavía en fase de estudio, por lo que en el presenta trabajo utilizaremos segmentación lineal para la estimación de los valores iniciales de los parámetros del modelo.

CAPITULO 5

EL SISTEMA BASICO DE RECONOCIMIENTO

5.1. INTRODUCCION

En este capítulo describiremos el diseño de un sistema de reconocimiento de palabras aisladas basado en modelos ocultos de Markov discretos y cuantización vectorial con distancia euclídea entre vectores de parámetros que representan características espectrales estáticas de la señal de voz.

Además, se discutirán y se aportarán resultados sobre diferentes técnicas de ponderación de los componentes de los vectores de parámetros, y sobre la incorporación de características adicionales tales como la energía logarítmica de la señal, así como características dinámicas del espectro de la señal, útiles para la caracterización de las zonas transicionales de los espectros, correspondientes a las coarticulaciones producidas entre los diferentes fonemas de las palabras.

Por último, se presentarán resultados relativos a la sintonización de parámetros y tasas de error para el sistema básico de reconocimiento de palabras aisladas, que se utilizarán más adelante como referencia para evaluar las mejoras introducidas por las modificaciones del sistema que se discutirán en los restantes capítulos.

En la sección 5.2 se describe el esquema general del sistema de reconocimiento, y se dan resultados previos sobre tasas de error cuando se emplea un vector de parámetros formado exclusivamente por coeficientes cepstrum de la señal.

En la sección 5.3 se revisan las aproximaciones utilizadas en la bibliografía para la ponderación de la distancia (medida de similitud espectral) entre vectores de coeficientes cepstrum, así como transformaciones de dichos coeficientes, para la mejora del rendimiento

del sistema. Se justificará de forma teórica la elección de una técnica de *liftering* frente a otras alternativas, y se mostrarán los resultados obtenidos para esta modificación de la función distancia.

En el apartado 5.4 se discute la incorporación de nuevos parámetros al vector de características. En concreto se discute la incorporación de la energía logarítmica de la señal, y características dinámicas del espectro. En cuanto a la integración de dichos parámetros al sistema, se discuten las dos alternativas propuestas en la bibliografía como son la utilización de una función distancia compuesta, manteniendo un diccionario VQ único, y la suposición de independencia estadística de observaciones, modificando las fórmulas de reestimación y evaluación para los modelos discretos de Markov a fin de manejar un conjunto de símbolos en lugar de uno único para representar cada observación

Por último, en la sección 5.5 se describirá la incorporación de la duración de los estados del modelo al sistema de reconocimiento. Se revisarán las diferentes aproximaciones propuestas en la bibliografía y en la sección 5.6 se mostrarán resultados finales para la configuración final del sistema básico de reconocimiento.

5.2. DESCRIPCION GENERAL DEL SISTEMA

A continuación presentamos un esquema general del sistema de reconocimiento. Este se basa en la modelización mediante modelos ocultos de Markov de un conjunto de palabras descrito en el capítulo 2, de forma que el reconocimiento se realiza en base a palabras aisladas, mediante la evaluación de las probabilidades *a posteriori* de generación de los diferentes modelos HMM, previamente construidos para cada palabra, por lo tanto, el sistema pertenece al conjunto denominado de reconocimiento de palabras aisladas.

En la figura 5.1 se muestra un diagrama de flujo de datos para el sistema. En el bloque de extracción de parámetros se incluye también la extracción de la energía logarítmica y de los parámetros dinámicos, aunque en determinadas fases del diseño sólo se tengan en cuenta parte de los parámetros que figuran en el diagrama. A continuación describimos cada una de las partes que componen el sistema.

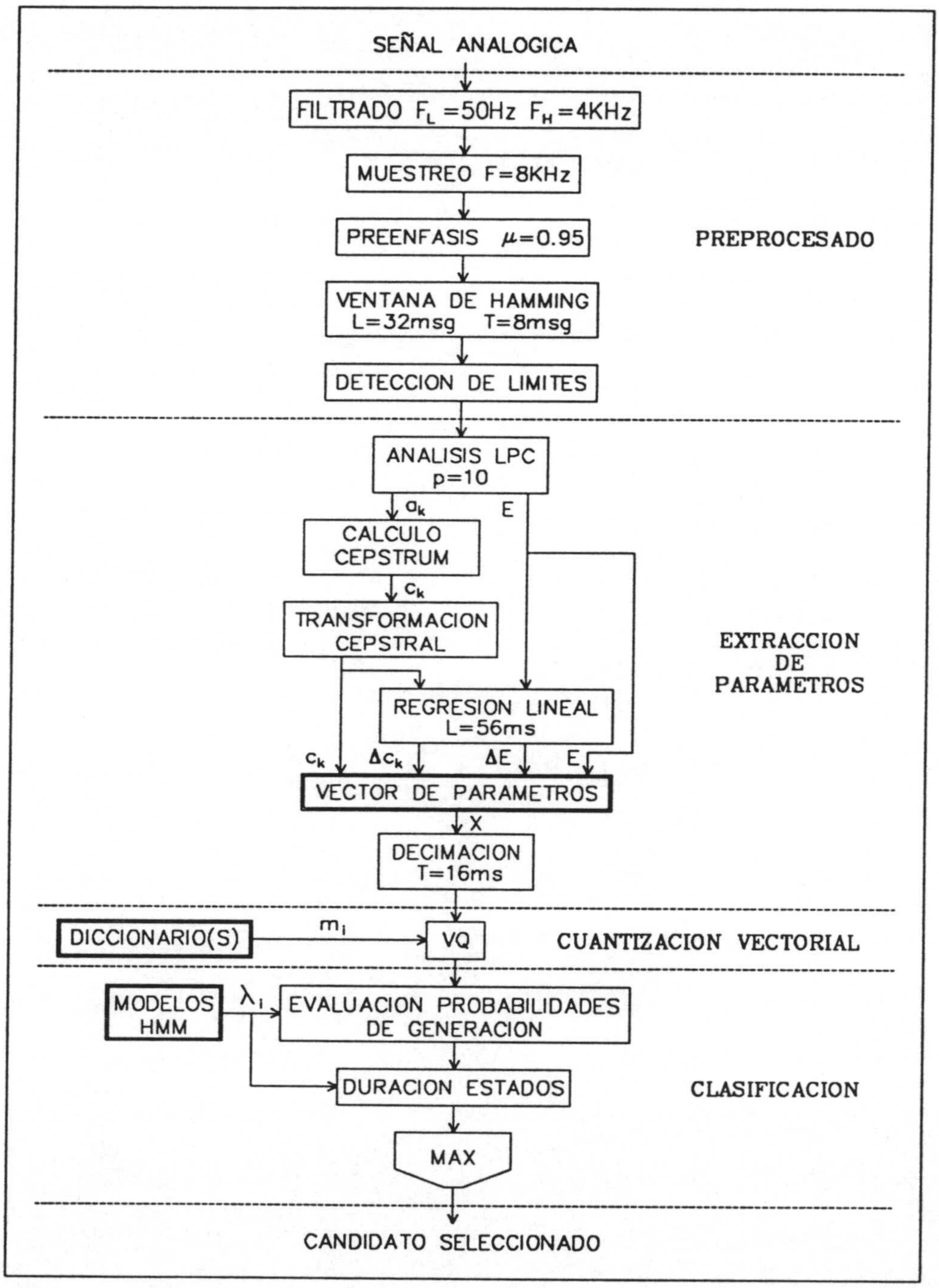

Figura 5.1. Esquema general del sistema básico de reconocimiento

5.2.1. PREPROCESADO DE LA SEÑAL

En este primer bloque del sistema, la señal de voz analógica es filtrada y muestreada adecuadamente para obtener una representación digital de la misma. En la implementación actual del sistema, la frecuencia de muestreo está fijada a 8 KHz, con un filtrado previo que limita la señal a la banda de frecuencias comprendida entre 60 Hz y 3800 Hz. Una vez muestreada la señal, se preenfatiza [Markel76b] con un filtro digital de función de transferencia

$$H_p(z) = 1 - \mu z^{-1} \quad ; \quad \mu = 0.95 \tag{5.1}$$

que elimina posibles niveles de contínua, y compensa la caida de 6 dB/década que presenta el espectro promedio de la voz humana. Esta caída es debida a los efectos de radiación y es conveniente eliminarla del proceso de análisis posterior.

Una vez hecho ésto, se extrae una serie consecutiva de segmentos de la señal obtenidos desplazando sobre ésta una ventana de Hamming de 32 milisegundos de duración en pasos de 8 milisegundos. Estos valores son apropiados para realizar un análisis LPC asíncrono de la señal de voz [Markel76b] con suficiente resolución temporal. Aunque estrictamente sólo sería necesario utilizar un muestreo temporal de periodo (desplazamiento de la ventana de Hamming) 16 milisegundos para evitar el *aliasing* de los espectros estimados para cada segmento [Rabiner78b], consideramos un periodo menor para obtener valores más suavizados de los parámetros dinámicos del espectro. Una vez finalizado el proceso de análisis, los parámetros obtenidos son decimados a un periodo de 16 milisegundos.

Sobre los segmentos obtenidos de esta forma, se evalúan las funciones energía y pasos por cero de la señal. Estas dos funciones se utilizan en un algoritmo explícito de delimitación [Segura84] similar al propuesto por Rabiner [Rabiner75a, Lamel81], para la obtención de los límites de las palabras, eliminando los silencios inicial y final con los que son grabadas. Este proceso de delimitación se describe con detalle en el capítulo 2.

5.2.2. EXTRACCION DE PARAMETROS

Sobre los segmentos obtenidos en la fase de preprocesado, tal y como se indica en la sección anterior, se realiza un análisis autoregresivo LPC en el que, a partir de los coeficientes de autocorrelación de los valores de la señal contenidos en el segmento considerado, se calculan los $p=10$ primeros coeficientes de predicción lineal. Para este cálculo se utiliza el algoritmo recursivo de Durbin [Rabiner78]. Junto con estos 10 coeficientes, también se extrae la energía logarítmica de las muestras de la señal contenidas en el segmento, simplemente sumando sus amplitudes al cuadrado y calculando el logaritmo natural del resultado.

Características espectrales estáticas

A partir de los coeficientes de predicción lineal, y utilizando la expresión (4.8), se extraen los 20 primeros coeficientes cepstrum correspondientes a la envolvente del espectro LPC del segmento de señal. Opcionalmente, estos coeficientes son transformados para obtener representaciones más adecuadas para tareas de reconocimiento. Estas transformaciones se detallarán más adelante.

De esta forma se obtiene un conjunto de 21 parámetros (20 cepstrum + la energía logarítmica), de entre los cuales se extraerán las utilizados para la representación de las características espectrales estáticas de los segmentos de señal.

Características espectrales dinámicas

A partir de los parámetros antes estimados, el sistema extrae un conjunto de parámetros para caracterizar la evolución temporal de las características espectrales de la señal. Esta caracterización se realiza en base a una estimación de la "velocidad" de variación de dichos parámetros, estimación que se realiza utilizando los valores del primer coeficiente de regresión lineal de los parámetros estáticos (cepstrum y energía) sobre un conjunto de segmentos en torno al actual. Los parámetros así obtenidos se denominan Δcepstrum y Δenergía, y se definen en la forma siguiente

$$\Delta c_t(n) = \frac{\sum_{m=-R}^{+R} m\, c_{t+m}(n)}{\sum_{m=-R}^{+R} m^2} \qquad (5.2a)$$

$$\Delta E_t = \frac{\sum_{m=-R}^{+R} m\, E_{t+m}}{\sum_{m=-R}^{+R} m^2} \qquad (5.2b)$$

donde el subíndice t indica el instante de tiempo correspondiente al segmento considerado, y R es la semilongitud del intervalo de regresión utilizado; E es la energía logarítmica y $c(n)$ es el n-ésimo coeficiente cepstrum de la señal.

Los valores así obtenidos son una aproximación a la pendiente de la evolución temporal de los parámetros estáticos o, equivalentemente, la velocidad de variación de dichos parámetros.

La longitud del intervalo de regresión debe elegirse lo suficientemente grande como para obtener estimaciones adecuadas de las características dinámicas del espectro, y lo suficientemente breve como para que no se introduzca un suavizado excesivo en los valores estimados, de forma que los parámetros dinámicos modelen adecuadamente las zonas transicionales de la señal producidas por los efectos coarticulatorios entre los fonemas de las palabras. En el presente trabajo elegimos un valor de 56 ms ($R = 3$ segmentos) para el intervalo de regresión, del mismo orden del utilizado en otros trabajos [Furui86a, Soong88b].

Aunque se pueden derivar coeficientes de regresión de orden superior como coeficientes de polinomios ortogonales de ordenes superiores [Appelbaum90], Furui [Furui81] mostró que, para la comparación de características dinámicas de espectros, el coeficiente de primer orden es usualmente suficiente.

En base a estos coeficientes, se puede estimar la variación temporal del espectro logarítmico de la señal en la forma

$$\frac{\partial F_t(e^{jw})}{\partial t} = \frac{\partial\left[\log H_t(e^{jw})\right]}{\partial t} \equiv \sum_{n=-\infty}^{+\infty} \Delta c_t(n)\, e^{-jwn} \tag{5.3}$$

donde el subíndice t denota la dependencia implícita de las magnitudes consideradas con el tiempo. Como se hizo para los espectros estáticos, se puede definir una aproximación a la distancia media entre las derivadas temporales de los logaritmos de las envolventes espectrales de forma análoga a la definida en (4.12)

$$d_{\Delta CEP}(x_r\ ,\ x_s) = \sum_{n=1}^{L} [\,\Delta c_r(n) - \Delta c_s(n)\,]^2 \tag{5.4}$$

que permite calcular similitudes entre derivadas temporales de dos envolventes espectrales logarítmicas. Como la expresión (4.12), (5.4) representa una aproximación a la distancia.

En la figura 5.2 se muestran los espectrogramas de una palabra generados a partir de los coeficientes cepstrum y Δcepstrum obtenidos en la fase de extracción de parámetros del sistema. Como puede comprobarse de la figura, las zonas transicionales entre fonemas quedan bien diferenciadas de las zonas estacionarias porque las primeras presentan valores significativos en el espectrograma dinámico de la señal a diferencia de las segundas.

En la bibliografía pueden encontrarse otras aproximaciones a la caracterización de las propiedades dinámicas de los espectros de la señal de voz, en las que se utiliza una aproximación en diferencias finitas para las derivadas [Lee88b]. Sin embargo, aunque esta aproximación es más sencilla de cálculo, la evolución temporal de los parámetros obtenidos mediante el método de regresión es más suave.

Por último, el vector resultante, de 42 componentes, formado por parámetros estáticos y dinámicos del espectro de la señal, es decimado a 16 ms promediando los valores de dos segmentos consecutivos de la señal, lo que reduce los requerimientos de cálculo en las etapas posteriores del sistema. Esta decimación no constituye pérdida de información ya que, debido al filtrado introducido por la ventana de Hamming de 32 ms de duración, el periodo mínimo de los parámetros obtenidos del análisis es de 16 ms.

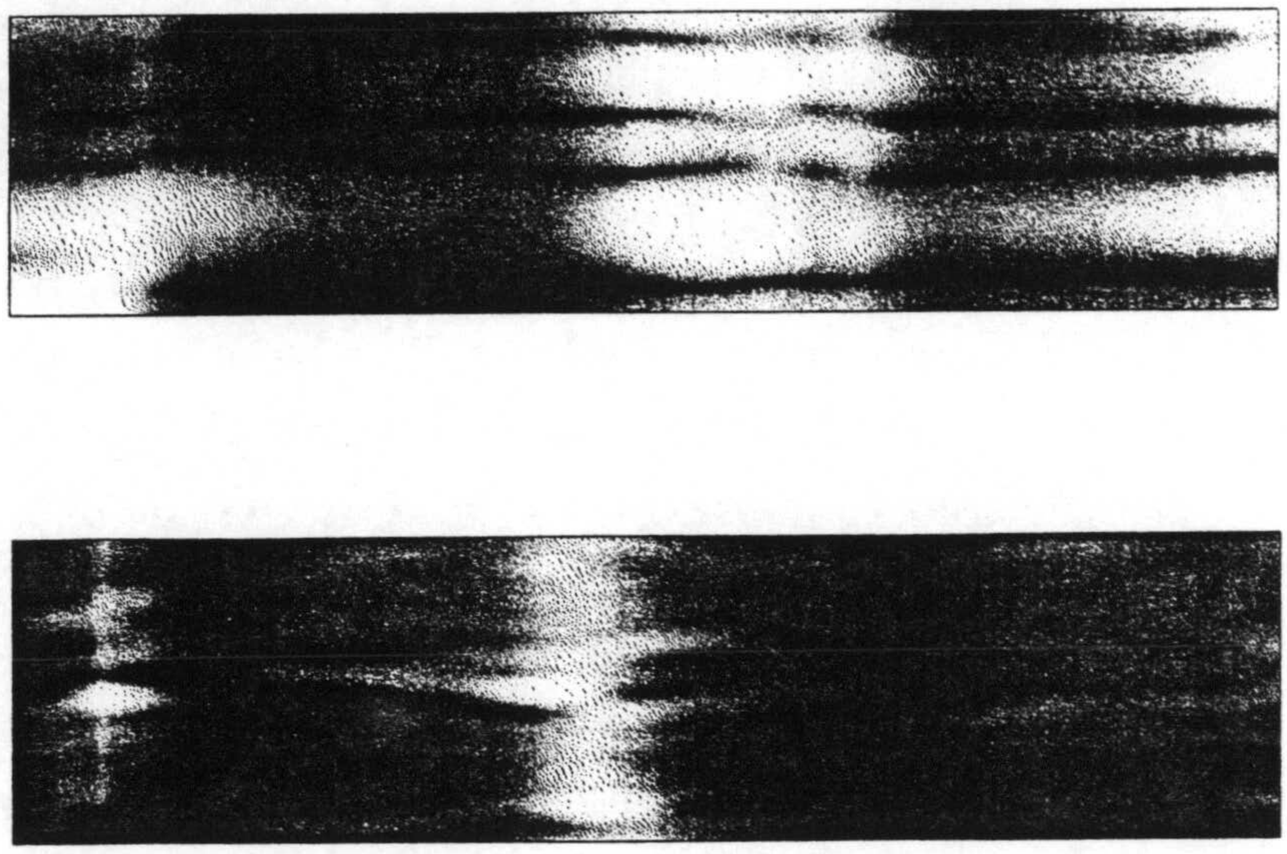

Figura 5.2. Espectrogramas de /SIETE/. Estático (arriba) y Dinámico (abajo)

5.2.3. CUANTIZACION VECTORIAL

Para el proceso de cuantización vectorial se utilizan uno o varios diccionarios construidos con el algoritmo jerárquico detallado en la sección 4.4. En el caso de utilizar un vector formado por un único tipo de parámetros (cepstrum o Δcepstrum), el diccionario se construye utilizando una distancia euclídea entre vectores.

Para la otra posible configuración, en la que se utiliza un vector compuesto por diferentes tipos de parámetros, se utiliza una distancia compuesta, definiéndola como una combinación lineal de los valores de las distancias euclídeas para cada tipo de componentes. Los factores de la combinación lineal se determinan experimentalmente. O bien se utilizan diccionarios independientes para cada tipo de parámetros. El proceso completo se detallará más adelante.

El proceso de cuantización vectorial sustituye cada vector de parámetros por uno, o varios símbolos, dependiendo de la utilización de un único diccionario, o diccionarios

separados para cada tipo de parámetros. Este proceso de sustitución se realiza en base a la selección del centro del diccionario correspondiente con menor distancia respecto al vector considerado. Las secuencias de símbolos así obtenidas se utilizan como observaciones de los modelos HMM discretos.

5.2.4. CLASIFICACION

Una vez transformadas las palabras en secuencias de símbolos a través de las fases de preprocesado, extracción de parámetros y cuantización vectorial, el proceso final de clasificación del sistema consiste en la evaluación de las probabilidades *a posteriori* de generación de los modelos de Markov previamente construidos en la fase de entrenamiento del sistema. La clasificación se realiza seleccionando el modelo cuya probabilidad de generación es máxima para la secuencias de símbolos observada.

Opcionalmente, el sistema incorpora al proceso de decisión, información relativa a las duraciones de los estados para la secuencia incógnita. Esto se hace obteniendo la decodificación en estados de la secuencia mediante el algoritmo de Viterbi, y determinando las duraciones de los estados obtenidos para esta decodificación. A continuación se evalua la probabilidad de estas duraciones utilizando las distribuciones de probabilidad de duración de estados para cada modelo, obtenidas en la fase de entrenamiento a partir de la decodificación en estados de cada una de las secuencias de entrenamiento para cada modelo. Esta probabilidad se combina con la probabilidad de generación del modelo para obtener una probabilidad final que es la utilizada en la clasificación de las secuencias incógnita. La descripción detallada de este proceso se mostrará más adelante.

5.2.5. ENTRENAMIENTO DEL SISTEMA

El entrenamiento del sistema se realiza en base a los vectores de parámetros obtenidos para un conjunto de repeticiones de cada palabra que denominaremos conjunto de entrenamiento. La totalidad de estos vectores (los de todas las repeticiones de todas las palabras en este conjunto) se utiliza como conjunto de vectores de entrenamiento para la construcción del diccionario.

Una vez construido el diccionario VQ, el conjunto de entrenamiento es cuantizado transformando las palabras, hasta ahora caracterizadas por secuencias de vectores, en

secuencias de símbolos discretos. Las secuencias de símbolos así obtenidas para cada palabra, se utilizan para la estimación de los parámetros de los modelos de Markov. El método utilizado en el entrenamiento de los modelos es un algoritmo iterativo que parte de los valores iniciales obtenidos mediante segmentación lineal de las secuencias de entrenamiento, y reestima los valores mediante el método Baum-Welch.

5.2.6. CONFIGURACION BASICA DEL SISTEMA. RESULTADOS PREVIOS

Como punto de partida en el diseño del sistema de reconocimiento, se realizó un experimento para determinar las tasas de error de un sistema básico en el que únicamente se utilizan coeficientes cepstrum en el vector de parámetros para la caracterización de las propiedades espectrales de los segmentos de señal obtenidos en el preprocesado.

Conjuntos de entrenamiento y test

Dado el volumen relativamente reducido de la base de datos, los experimentos realizados sobre la misma y a los que se refieren los resultados que se mostrarán en lo que sigue, salvo que se indique lo contrario, se diseñaron en base a una técnica similar a la denominada *leaving one out* [Duda73]. Cuando el volumen de la base de datos es lo sificientemente grande, el error de un sistema de reconocimiento puede estimarse sin más que divir ésta en dos conjuntos disjuntos, uno de los cuales se utiliza como conjunto de entrenamiento del sistema, y el otro para el test del mismo. Sin embargo, cuando el volumen de la base de datos es reducido, la elección de un conjunto de test grande, para que los resultados sean estadísticamente significativos, fuerza a utilizar un conjunto de entrenamiento reducido, con lo que el sistema estará pobremente entrenado. La elección de un conjunto de entrenamiento grande, deja pocos datos para el conjunto de test, lo que incidirá negativamente en la significación estadística de los resultados obtenidos.

En esencia, la aproximación *leaving one out* se basa en el hecho de que se puede realizar más de una partición disjunta de una base de datos. En el límite, se puede realizar tantas particiones disjuntas como secuencias forman la base de datos, de forma que en cada una de ellas sólo una secuencia forma parte del conjunto de test, y el resto forma parte del conjunto de entrenamiento. De esta forma, el sistema está virtualmente entrenado con todas las secuencias de la base de datos. Para obtener resultados significativos, basta con entrenar y evaluar el sistema sobre todas las particiones posibles de la base de datos, y promediar

los resultados. En el presente trabajo utilizaremos dos conjuntos de particiones de la base de datos, que pasamos a describir a continuación.

Experimentos nultilocutor L1OUT

El primer tipo de experimentos corresponde a conjunto de particiones para configuraciones multilocutor del sistema. En este tipo de experimentos, el sistema es evaluado sobre un conjunto de secuencias correspondientes a locutores que forman parte del conjunto de entrenamiento. Este tipo de experimentos los denominaremos en adelante L1OUT, y se realizan sobre tres particiones de la base de datos, en cada una de las cuales, dos repeticiones de cada palabra de la base de datos, pronunciadas por cada locutor se utilizan como conjunto de entrenamiento, y la tercera se utiliza como conjunto de test. Los resultados se obtienen promediando los correspondientes a cada una de las tres particiones. El número de repeticiones de cada palabra en el conjunto de entrenamiento es de 80, y el sistema se evalua sobre un total de 1920 casos correspondientes a las tres repeticiones de cada palabra para cada uno de los 40 locutores.

Experimentos independientes del locutor L4OUT

El segundo tipo de experimentos realizados corresponde a una configuración independiente del locutor del sistema. En esta configuración, el sistema es evaluado sobre un conjunto de locutores diferentes a los que componen el conjunto de entrenamiento. Para estos experimentos se han utilizado cinco particiones de la base de datos, en cada una de las cuales, las tres repeticiones de cada palabra pronunciadas por ocho locutores (cuatro masculinos y cuatro femeninos) forman el conjunto de test, y el resto (32 locutores, 16 masculinos y 16 femeninos) forman el conjunto de entrenamiento. El número de repeticiones de cada palabra en el conjunto de entrenamiento es de 96, y de nuevo, el sistema se evalua sobre 1920 casos correspondientes a los 40 locutores de la base de datos, con un conjunto de entrenamiento formado por 32 locutores.

Construcción de los diccionarios

Para la construcción de los diccionarios se utilizó el algoritmo jerárquico expuesto en la sección 4.4 con una distancia euclídea entre vectores formados por los 10 primeros coeficientes cepstrum de los segmentos de señal. El número de centros para el diccionario se fijó a 64, con un parámetro de convergencia V_r del 0.1% para la distorsión media del diccionario, y un valor μ=0.001 para la perturbación de los centros en el proceso de división jerárquica. El conjunto de vectores utilizados en la construcción del diccionario es el formado por todos los vectores de las secuencias que componen el conjunto de entrenamiento.

Entrenamiento de los modelos HMM

En el entrenamiento de los modelos se utilizó una formulación de estimación de máxima probabilidad, de forma que el modelo de cada palabra del vocabulario se entrenó con repeticiones de dicha palabra e independientemente de los modelos correspondientes al resto de las palabras del vocabulario.

Para la estimación de los parámetros de los modelos HMM se utilizaron las fórmulas de reestimación Baum-Welch en un proceso iterativo de estimación-modificación, en el que, a partir de las estimaciones iniciales de los parámetros, obtenidas mediante una segmentación lineal de las secuencias de entrenamiento, se evalúan las probabilidades hacia delante y hacia atrás, y con éstas se reestiman los valores de los parámetros.

Este proceso se repite iterativamente hasta que la probabilidad de generación del conjunto de secuencias de entrenamiento para cada modelo no varía de forma apreciable. La decisión de finalización del algoritmo se realiza en base a la variación relativa de la probabilidad logarítmica de generación, de forma que se itera hasta que ésta no rebasa un umbral prefijado. En estos experimentos se fijó al 0.1% dicho umbral.

La suavización de las probabilidades de generación de símbolos se realizó fijando un límite inferior de 10^{-4}, de forma que, tras cada iteración de reestimación, todas las probabilidades inferiores a dicho valor se sustituyeron por éste, normalizando después los valores a la unidad.

El número de estados para todos los modelos se fijó a 5, dado que es un valor próximo al número medio de fonemas que forman las palabras del vocabulario. Al final del capítulo se muestran resultados relativos a la determinación experimental del valor óptimo de dicho parámetro.

En los resultados que se muestran a continuación, no se utilizó ninguna transformación en los coeficientes cepstrum, ni tampoco se incorporó información sobre la duración de los estados de los modelos.

Resultados previos de reconocimiento

Para el sistema de reconocimiento antes propuesto, se evaluaron las tasas de error sobre la configuraciones multilocutor (experimento L1OUT) y independiente del locutor (experimento L4OUT) obteniendo los resultados que se muestran en la Tabla I

CONFIGURACION	ERRORES
L1OUT	15.07%
L4OUT	18.90%

Tabla I. Resultados previos para el sistema de referencia

Como se puede observar, los resultados no son satisfactorios. Además existe una amplia diferencia (en torno al 4%) entre los obtenidos para las dos configuraciones del sistema, siendo el error especialmente alto para el caso independiente del locutor, en el que el sistema no está entrenado con los locutores del conjunto de test. Estos primeros resultados sugieren el hecho de que los parámetros utilizados no son adecuados para la caracterización de las observaciones en el caso de múltiples locutores (tanto multilocutor como independiente del locutor).

En las siguientes secciones se describen modificaciones sobre el conjunto de parámetros utilizados para mejorar el rendimiento del sistema. Estas modificaciones se centrarán en la transformación de los coeficientes cepstrum, y en la incorporación de nuevas carcterísticas del espectro de la señal al vector de parámetros.

5.3. TRANSFORMACIONES DE LOS COEFICIENTES CEPSTRUM

La elección de la función distancia utilizada en un sistema de reconocimiento basado en comparación de patrones o cuantización vectorial, es una cuestión de gran importacia, ya que gran parte de tales sistemas descansa en el cálculo de distancias entre segmentos de señal. Por este motivo, el estudio del comportamiento de dichos sistemas frente a las medidas de distancia utilizadas ha sido un problema ampliamente estudiado.

Se ha propuesto una gran variedad de medidas de distancia [Itakura75a, Gray76], entre las que la razón de semejanza (*Likelihood Ratio*), y la distancia euclídea entre vectores de coeficientes cepstrum han sido las más ampliamente utilizadas. La distancia euclídea entre coeficientes cepstrum ha sido utilizada con diferentes variantes, debido a su forma sencilla, y a la propiedad de que es una aproximación a la distancia cuadrática media entre los espectros logarítmicos representados por los coeficientes cepstrum, tal y como se mostró en la sección 4.4.1.

Una de estas variantes es la distancia cepstral con pesado estadístico, inicialmente propuesta por Furui [Furui81], y que consiste en pesar cada componente del cepstrum con la inversa de su varianza estadística.

Paliwal [Paliwal82] estudió el comportamiento de una distancia cepstral pesada en la que cada componente del vector de coeficientes cepstrum se pesa con el índice del coeficiente.

Juang [Juang87] abordó el problema genérico del pesado de la distancia cepstral como un problema de filtrado en el dominio del cepstrum, y probó varias formas para el filtro cepstral.

Todos estos métodos de pesado de la distancia cepstral han ofrecido mejoras significativas en el rendimiento de los sistemas en que han sido probadas.

Una alternativa al pesado de la distancia euclídea, es la de definir dicha distancia sobre espectros muestreados en escala MEL. La escala MEL es una escala logarítmica de frecuencias que aproxima la escala perceptual del oido humano. Lee [Lee88b] mostró que una transformación de frecuencia obtenida a través de la transformación bilineal

[Oppenheim75], y que aproxima la escala de frecuencias MEL, ofrece mejoras significativas en el rendimiento de su sistema de reconocimiento.

A continuación describiremos con detalle estas aproximaciones a la modificación de los coeficientes cepstrum, y aportaremos argumentos que justifican la elección del método de filtrado cepstral frente a los demás.

5.3.1. PESADO ESTADISTICO

Las distribuciones estadísticas de los valores de los coeficientes cepstrum, estimadas para un conjunto amplio de locutores, muestran un comportamiento de tipo gausiano. Además, las varianzas decaen monótonamente con el índice del coeficiente [Tohkura87], así, los coeficientes de orden alto tienen una varianza mucho menor que los coeficientes de orden bajo, lo que implica que estos últimos representan una fracción menor de la distancia total que los primeros.

Una distancia con mayor significado estadístico, utilizada con frecuencia en problemas de clasificación de patrones, es la distancia de Mahalanobis, que se puede definir en la forma

$$d_{MCEP} = (c_r - c_s)\,\Sigma^{-1}(c_r - c_s)^t \qquad (5.5)$$

donde c_r y c_s son dos vectores de coeficientes cepstrum y Σ es la matriz de covarianza de dichos coeficientes. Los principales inconvenientes de esta distancia son, la gran cantidad de cálculo requerida, y la estimación de los coeficientes fuera de la diagonal de Σ^{-1} debido a su bajo valor y al proceso de inversión de la matriz, lo que generalmente lleva asociados errores de estimación.

Una posible solución a estos problemas es la de asumir que la matriz de covarianza es diagonal, y redefinir consecuentemente la distancia en la forma

$$d_{MCEP} = \sum_{n=1}^{L} w(n)[c_r(n) - c_s(n)]^2 \tag{5.6a}$$

$$= \sum_{n=1}^{L} [\hat{c}_r(n) - \hat{c}_s(n)]^2 \tag{5.6b}$$

$$\hat{c}(n) = \frac{c(n)}{\sigma_n} \tag{5.6c}$$

donde $w(n)$ es la inversa de la varianza del coeficiente $c(n)$. De esta forma se obtiene una distancia entre coeficientes ecualizados en varianza $\hat{c}(n)$. σ_n es la desviación típica del coeficiente $c(n)$.

5.3.2. DISTANCIA PESADA

Otra forma de pesado de la distancia entre vectores de coeficientes cepstrum, fue inicialmente propuesta por Paliwal [Paliwal82], basada en consideraciones perceptuales, y se define en la forma

$$d_{QCEP} = \sum_{n=1}^{L} n^2 [c_r(n) - c_s(n)]^2 \tag{5.7a}$$

$$= \sum_{n=1}^{L} [\hat{c}_r(n) - \hat{c}_s(n)]^2 \tag{5.7b}$$

$$\hat{c}(n) = n\, c(n) \tag{5.7c}$$

Esta distancia fue comparada por Tohkura [Tohkura87] con la obtenida mediante pesado estadístico, obteniendo resultados similares. De hecho, estas dos distancias presentan grandes similitudes si se tiene en cuenta [Oppenheim75a, Juang87] que el decaimiento de los coeficientes cepstrum de una señal el del orden de $1/k$ donde k es el índice del coeficiente.

Esta distancia está relacionada con la distancia entre las pendientes de los espectros logarítmicos. Si derivamos (4.9) respecto a la frecuencia obtenemos

$$F'(e^{jw}) = \frac{\partial F(e^{jw})}{\partial w} = \sum_{n=-\infty}^{+\infty} -j\, n\, c(n)\, e^{jwn} \tag{5.8}$$

y por lo tanto, la distancia entre las pendientes espectrales se puede escribir en la forma siguiente

$$d[F'_r(e^{jw}), F'_s(e^{jw})] = \sum_{n=-\infty}^{+\infty} n^2 [c_r(n) - c_s(n)]^2 \tag{5.9}$$

de donde se puede deducir que la distancia definida por (5.7) es una aproximación a la distancia entre las pendientes de los espectros logarítmicos.

5.3.3. FILTRADO CEPSTRAL

Tohkura [Tohkura87] mostró que la mejora introducida por el pesado estadístico de la distancia cepstral d_{MCEP} se debe al deénfasis de los coeficientes de orden bajo, y no al énfasis de los coeficientes de orden alto del cepstrum de la señal. Este efecto es similar para la distancia cepstral pesada d_{QCEP}. Además comprobó que el rendimiento de la distancia (en términos de un menor error de reconocimiento) aumentaba al aumentar el número de coeficientes cepstrum considerado en la misma, pero sólo hasta llegar a un número de coeficientes igual al orden de predicción utilizado en el análisis LPC. A partir de éste valor, vuelve a aumentar el error del sistema de reconocimiento. Estos dos hechos inducen a pensar que el énfasis excesivo de los coeficientes de orden alto del cepstrum provoca efectos perjudiciales, y que la principal ventaja del pesado de la distancia radica en el deénfasis de los coeficientes de orden bajo.

Juang [Juang87] realizó un estudio sobre las fuentes de la variabilidad de los coeficientes cepstrum extraidos de análisis LPC. En éste se concluye que, tanto los coeficientes cepstrum de orden bajo como los de orden alto, presentan variabilidades no deseadas para procesos de comparación de patrones. De un lado, los coeficientes de orden bajo (típicamente los 2 ó 3 primeros) presentan una variabilidad provocada por las condiciones del canal de transmisión utilizado en la adquisición de la señal y caractrísticas propias del locutor tales como la forma de la onda glotal. Estas contribuciones afectan principalmente al decaimiento global del espectro, que a su vez se manifiesta principalmente en los primeros coeficientes del cepstrum de la señal.

En cuanto a los coeficientes de orden alto, éstos presentan una variabilidad artificialmente alta debido al proceso de análisis, lo que se puso de manifiesto mediante estudios de simulación [Juang87].

Del conjunto de las consideraciones antes expuestas, se deduce que la función peso para la distancia entre vectores de coeficientes cepstrum debe deenfatizar tanto los coeficientes de orden bajo como los de orden alto. El proceso de pesado de los coeficientes cepstrum puede contemplarse como un filtrado en el dominio del cepstrum en la forma

$$d_{MCEP} = \sum_{n=1}^{L} w^2(n)[c_r(n) - c_s(n)]^2 \tag{5.10a}$$

$$= \sum_{n=1}^{L} [\hat{c}_r(n) - \hat{c}_s(n)]^2 \tag{5.10b}$$

$$\hat{c}(n) = w(n)\,c(n) \tag{5.10c}$$

donde $w(n)$ es una ventana en el dominio del cepstrum, que realiza un filtrado pasa-banda del espectro logarítmico de la señal en el dominio del cepstrum, y L es la longitud del filtro. En la figura 5.3 se representan algunas formas posibles para el filtro pasa-banda (o filtro de *liftering*). La figura 5.3a corresponde a una distancia euclídea sobre las secuencias truncadas de coeficientes cepstrum, la 5.3b corresponde a la distancia d_{QCEP} definida en (5.7), la 5.3c es una versión saturada de ésta última, y las 5.3 d e y f representan diferentes formas de filtros simétricos.

Experimentos sobre reconocimiento con distancias euclídeas sobre coeficientes cepstrum filtrados con algunas de estas ventanas mostraron [Juang87] que la que mejor tasa de reconocimiento ofrece es la correspondiente a un filtro seno-remontado tal y como el de la figura 5.3f de la forma

$$w(n) = 1 + \frac{L}{2}\,\mathrm{sen}\left(\frac{n\pi}{L}\right) \tag{5.11}$$

donde la elección de la longitud L de la ventana de *lifter* determina el número de coeficientes cepstrum considerado. En los experimentos antes citados de Juang, los resultados óptimos se obtuvieron para el valor L=12 en un sistema en el que el análisis

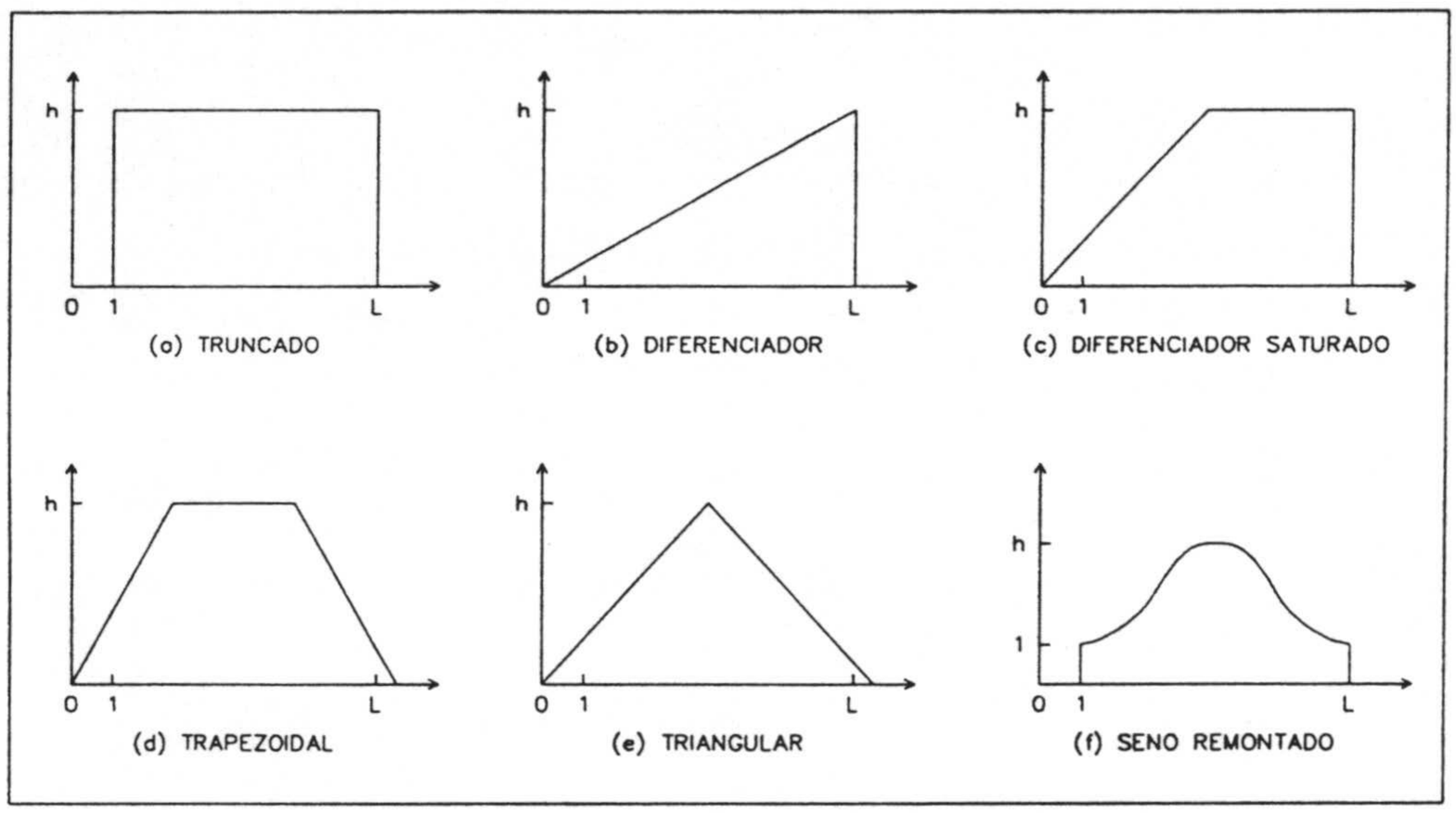

Figura 5.3. Filtros de liftering

LPC se realizó con orden de predicción p=8 y frecuencia de muestreo f_s = 6.6 KHz.

En cuanto al efecto de dicho filtrado sobre el espectro de la señal, en la figura 5.4a se muestran los espectros LPC correspondientes a siete segmentos consecutivos de una vocal, y en las figuras 5.4b y 5.4c se muestran los espectros correspondientes para los filtros mostrados en las figuras 5.3b y 5.3f. La señal se muestreó a 8 KHz y se realizó un

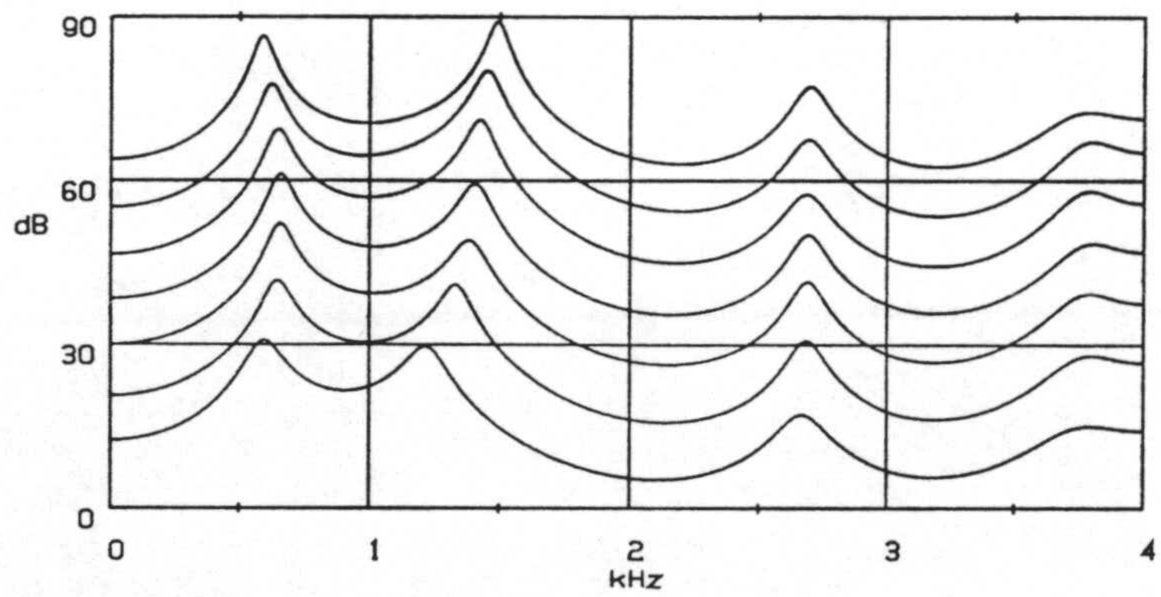

Figura 5.4a. Espectros originales LPC

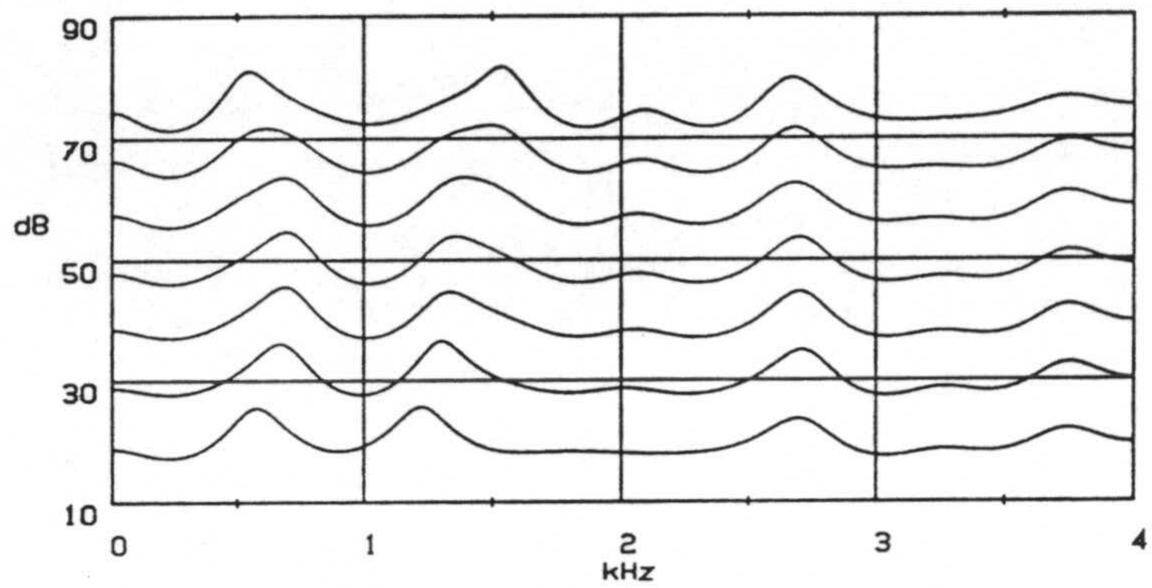

Figura 5.4b. Filtrado con diferenciador truncado

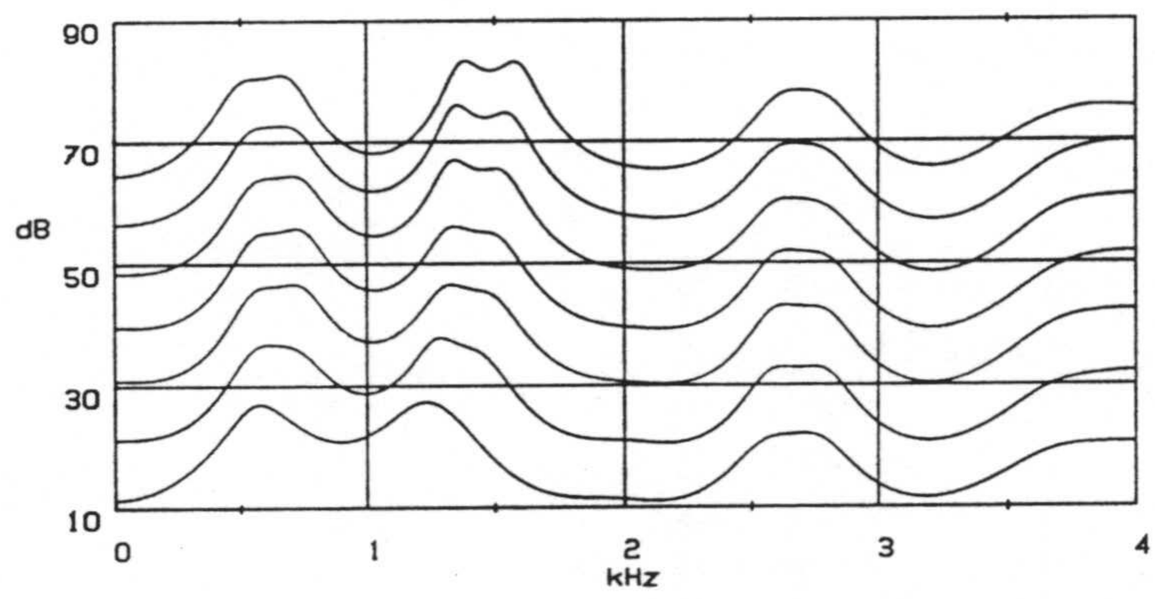

Figura 5.4c. Filtrado con seno remontado

análisis LPC con orden de predicción $p = 10$, a partir del cual se extrajeron los coeficientes cepstrum. La longitud de los filtros utilizados para la generación de los espectros de las figuras 5.4b y 5.4c fué L=16.

Como puede apreciarse, el efecto principal del filtrado es el "allanamiento" del espectro, y el "ensanchamiento" de los picos del mismo (correspondientes a los formantes), lo que reduce la excesiva sensibilidad de la distancia a las frecuencias formantes, preservando las características globales del espectro.

5.3.4. TRANSFORMACION DE LA ESCALA DE FRECUENCIAS

Otra aproximación posible a la modificación de la distancia cepstral es la propuesta por Davis y Mermelstein [Davis80], que mostraron que los coeficientes cepstrum en escala MEL dan los mejores resultados sobre un conjunto de parámetros que contiene los coeficientes cepstrum derivados en escala MEL (MFCC), coeficientes cepstrum en escala lineal a partir de FFT (LFCC), coeficientes cepstrum en escala lineal a partir de LPC (LPCC), coeficientes LPC (LPC) y coeficientes de reflexión (RC). Para los coeficientes cepstrum y los coeficientes de reflexión se utilizó una distancia euclídea y para los coeficientes LPC la distancia de Itakura-Saito. Las pruebas se realizaron en un sistema basado en programación dinámica DTW, monolocutor, sobre un conjunto de 52 palabras monosílabas. Los resultados de reconocimiento se reproducen en la Tabla II.

CONFIGURACION		ACIERTOS		
PARAMETRO	**DISTANCIA**	**LOCUTOR 1**	**LOCUTOR 2**	**MEDIA**
MFCC	EUCLIDEA	96.5%	95.0%	95.75%
LFCC	EUCLIDEA	94.7%	87.6%	91.15%
LPCC	EUCLIDEA	92.6%	87.3%	89.95%
LPC	ITAKURA	85.2%	84.3%	84.75%
RC	EUCLIDEA	83.1%	77.5%	80.30%

Tabla II. Comparación de diferentes parámetros.

En la tabla se observa el mejor rendimiento de los coeficientes cepstrum distribuidos en escala MEL frente a los demás parámetros. Además, se observa que los coeficientes cepstrum obtenidos a partir de FFT y LPC ofrecen unos resultados similares y superiores a los obtenidos con los coeficientes LPC con la distancia de Itakura y con los coeficientes de reflexión con distancia euclídea.

Escala MEL

La escala MEL es una aproximación a la escala logarítmica de percepción del oido humano, dada por las relaciones

$$f = 600 \operatorname{senh}(g/6) \tag{5.12a}$$

$$g = 6 \log\left[(f/600) + \sqrt{1 + (f/600)^2} \right] \tag{5.12b}$$

donde f está en Hertzios (escala lineal) y g en Barks (escala MEL). La correspondencia entre las dos escalas se muestra en la figura 5.5. Una distribución lineal de frecuencias en escala MEL corresponde a una distribución logarítmica de frecuencias en escala lineal, y por lo tanto, a realizar un muestreo mas frecuente en la zona de bajas frecuencias que en la zona de altas frecuencias. La obtención del espectro de una señal en escala MEL requiere del muestreo uniforme del mismo sobre la escala MEL, lo que puede realizarse a través de

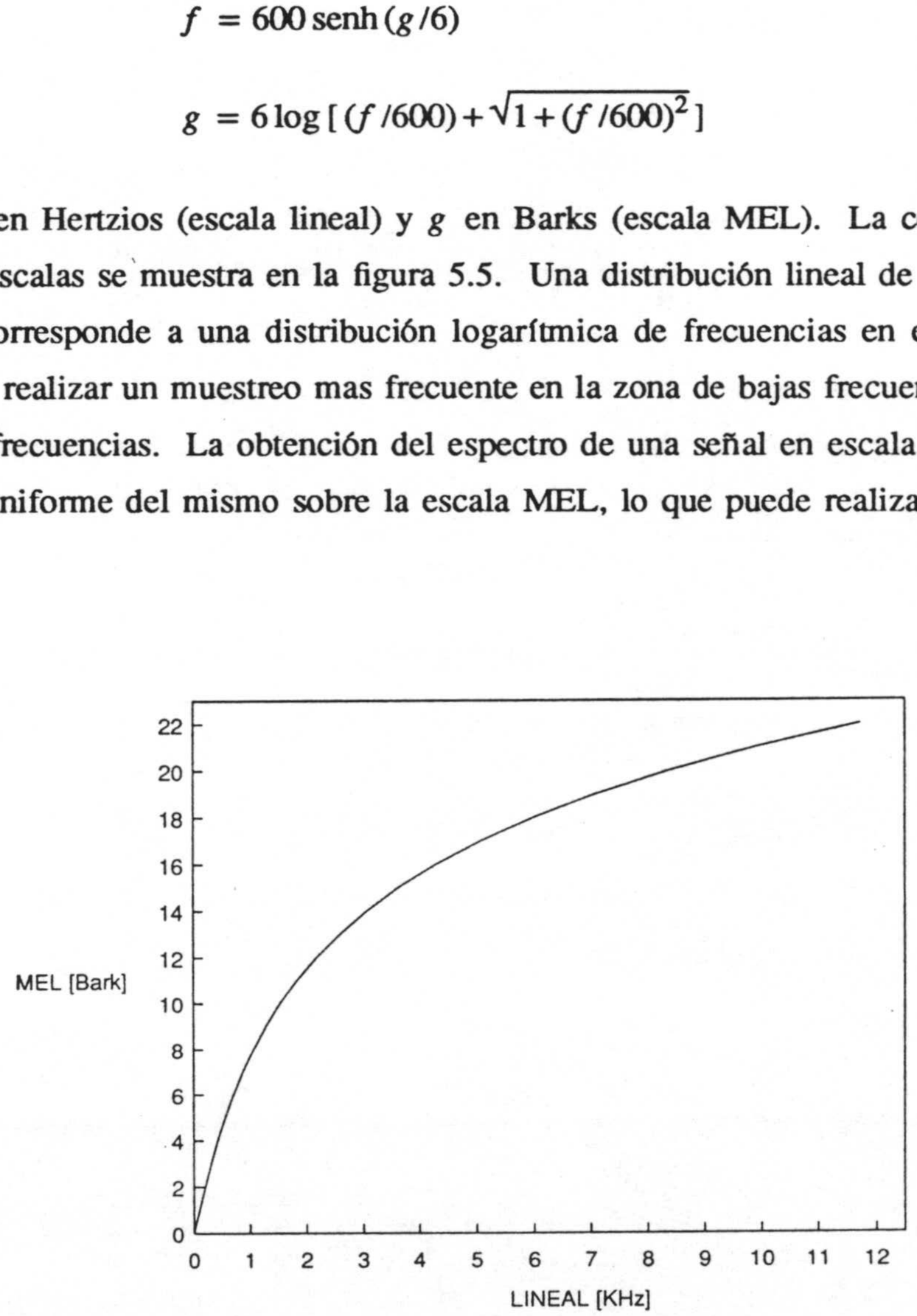

Figura 5.5. Correspondencia entre las escalas MEL y LINEAL

un banco de filtros pasa-banda con anchos de banda distribuidos uniformemente en escala MEL, sobre el rango de frecuencias deseado. A partir de estas muestras del espectro, se pueden obtener los coeficientes cepstrum mediante la transformada inversa de Fourier. En la figura 5.6 se muestran los diagramas de bloques del proceso para la obtención de los coeficientes cepstrum en escala MEL a partir de la señal $x(n)$ y a partir de los coeficientes cepstrum $c(n)$ obtenidos en escala lineal donde los valores $\hat{X}(k)$ y $\hat{c}(n)$ representan valores en escala MEL. Un método como el descrito fue utilizado por Davis [Davis80] para la extracción de coeficientes cepstrum en escala MEL a partir de espectros FFT, utilizando un banco de filtros triangulares con anchos de banda en escala MEL similar al mostrado en la figura 5.7.

Transformación bilineal

Una alternativa al proceso de transformación de los coeficientes cepstrum a escala MEL antes expuesto, fue propuesto por Shikano y utilizada por Lee [Lee88b], basado en la transformación bilineal para aproximar la escala de frecuencias MEL.

La transformación bilineal [Oppenheim75b] es una transformación definida sobre el plano complejo z, que realiza una transformación no lineal del eje de frecuencias. Esta

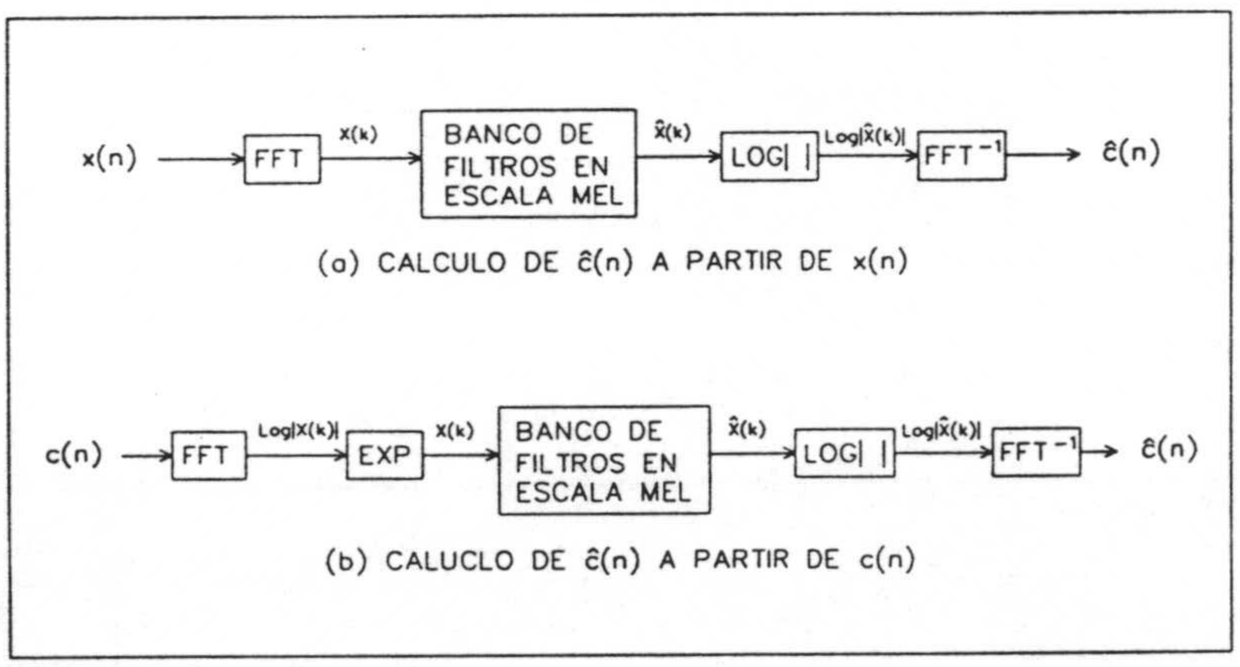

Figura 5.6. Obtención de los coeficientes cepstrum en escala MEL

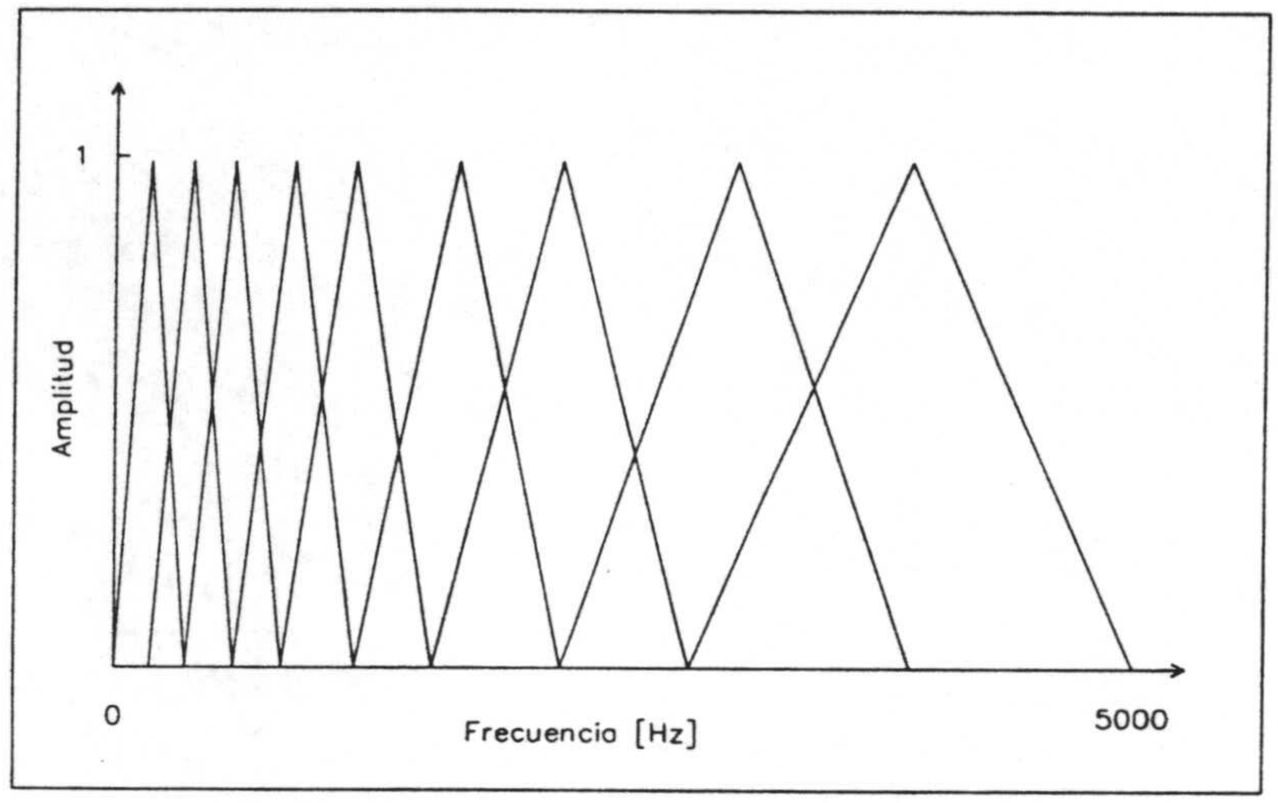

Figura 5.7. Banco de 9 filtros en escala MEL

transformación ha sido utilizada en el diseño de filtros digitales a partir de prototipos pasa-baja mediante un cambio en la escala de frecuencias.

En esencia, la transformación bilineal es un filtro pasa-todo que convierte el círculo unidad del plano complejo z en otro círculo unidad de un nuevo plano complejo Z de forma que la correspondencia angular entre los dos planos es no-lineal. La transformación bilineal tiene la forma siguiente

$$Z^{-1} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}} \quad ; \quad |\alpha| < 1 \tag{5.13}$$

donde z y Z son las variables complejas original y transformada respectivamente, y α es un parámetro que controla la transformación. Para obtener la relación entre las frecuencias, basta con evaluar la transformación sobre el círculo unidad, es decir

$$\begin{aligned} z &= e^{j\phi} \\ Z &= e^{jw} \end{aligned} \tag{5.14}$$

donde ϕ es la frecuencia original y w la transformada. Luego la transformación bilineal sobre el círculo unidad toma la forma

$$e^{-jw} = \frac{e^{-j\phi} - \alpha}{1 - \alpha e^{-j\phi}} \tag{5.15a}$$

$$e^{-j\phi} = \frac{e^{-jw} + \alpha}{1 + \alpha e^{-jw}} \tag{5.15b}$$

relaciones a partir de las que, después de ciertas transformaciones, se llega a la relación

$$\text{tg}\, w = \frac{(1-\alpha^2)\,\text{sen}\,\phi}{-2\alpha + (1+\alpha^2)\cos\phi} \tag{5.16}$$

La compresión-expansión realizada por la transformación bilineal está controlada por el valor del parámetro α. Tal y como se muestra en la figura 5.8, valores positivos de α provocan una compresión en la zona baja de frecuencias y una expansión en la zona alta, y para valores de α negativos se produce el efecto contrario. Puede comprobarse que, para frecuencia de muestreo de 8KHz, la transformación bilineal con $\alpha=0.4$ corresponde aproximadamente a la transformación de frecuencia a escala MEL, como puede apreciarse

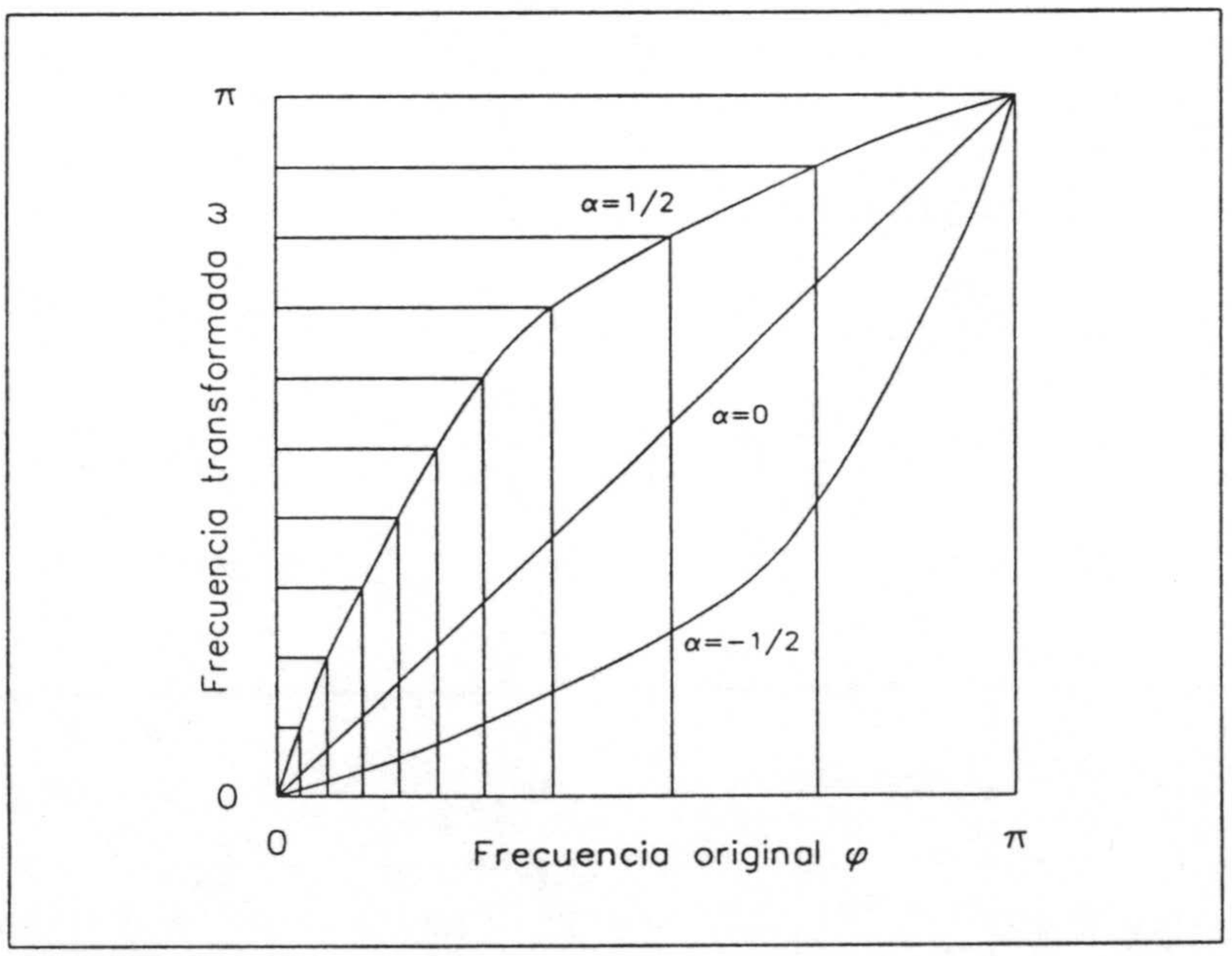

Figura 5.8. Correspondencia de frecuencias en la transformación bilineal

en la figura 5.9.

La ventaja de la utilización de la transformación bilineal en lugar de la transformación directa expuesta anteriormente es que permite obtener una expresión matricial para la transformación de los coeficientes cepstrum, que se aproximarán a los obtenidos mediante el cambio a escala MEL. La derivación completa de la matriz de transformación se muestra en el apéndice A, aquí sólo describiremos las relaciones principales.

Dado un conjunto de coeficientes cepstrum $c(k)$, podemos expresar su espectro logarítmico en la forma

$$F(e^{j\phi}) = \sum_{k=-\infty}^{+\infty} c(k)\, e^{-jk\phi} \tag{5.17}$$

Si denominamos $F(e^{jw})$ al espectro logarítmico transformado, podemos escribir, haciendo uso de (5.15a)

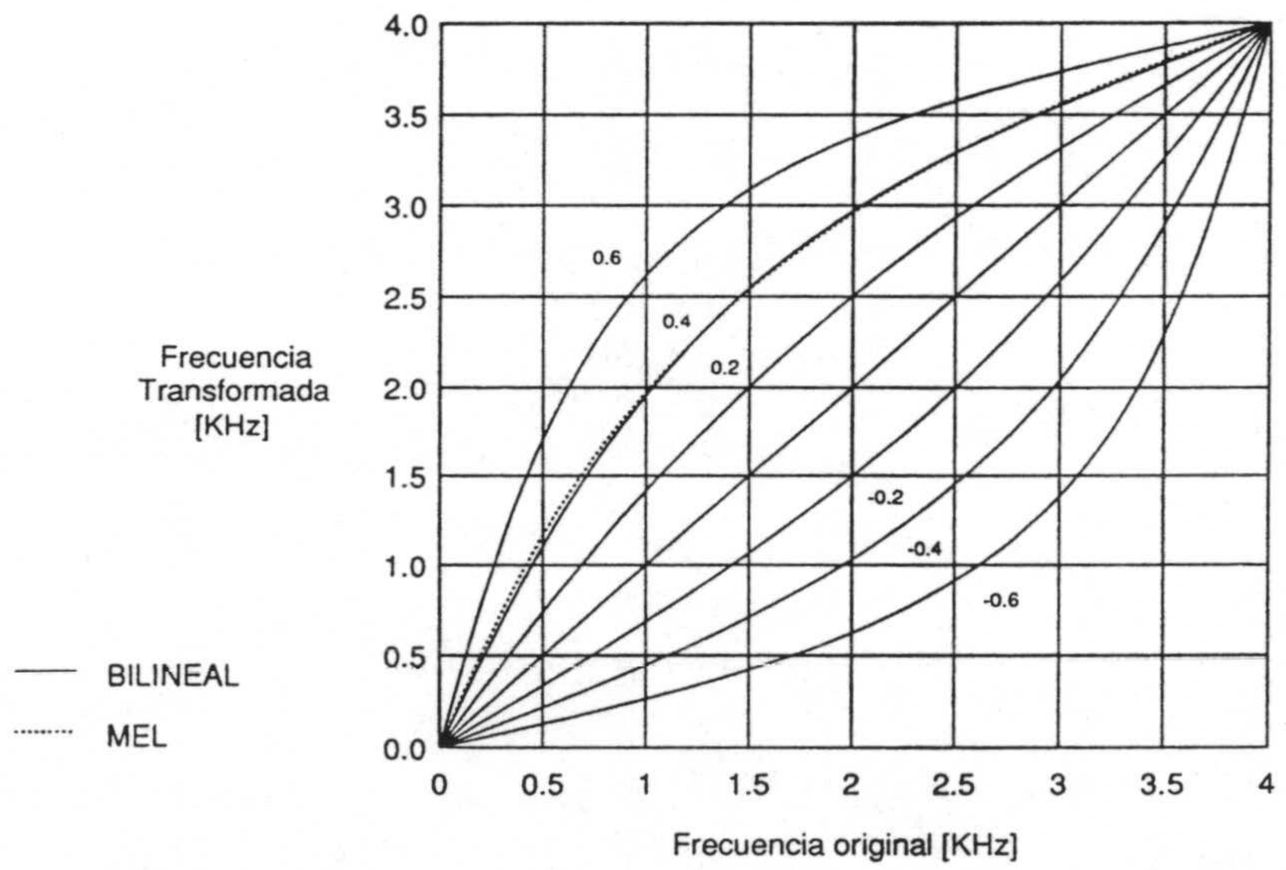

Figura 5.9. Comparación entre las escalas MEL y BILINEAL

$$F(e^{jw}) = \sum_{k=-\infty}^{+\infty} c(k) \left[\frac{e^{-jw} + \alpha}{1 + \alpha e^{-jw}} \right]^k \tag{5.18}$$

y los coeficientes cepstrum $\hat{c}(n)$ transformados pueden obtenerse calculando la transformada inversa de (5.18)

$$\hat{c}(n) = \int_{-\pi}^{+\pi} F(e^{jw}) e^{jnw} \frac{dw}{2\pi} \tag{5.19a}$$

$$= \sum_{k=-\infty}^{+\infty} c(k) \int_{-\pi}^{+\pi} \left[\frac{e^{-jw} + \alpha}{1 + \alpha e^{-jw}} \right]^k e^{jnw} \frac{dw}{2\pi} \tag{5.19b}$$

$$\hat{c}(n) = \sum_{k=-\infty}^{+\infty} c(n) W(n,k) \tag{5.20a}$$

$$W(n,k) = \int_{-\pi}^{+\pi} \left[\frac{e^{-jw} + \alpha}{1 + \alpha e^{-jw}} \right]^k e^{jnw} \frac{dw}{2\pi} \tag{5.20b}$$

Tal y como se muestra en (5.20), la transformación entre los dos conjuntos de coeficientes viene dada por la matriz de transformación lineal W. Los coeficientes de la transformación $W(n,k)$ pueden obtenerse mediante un algoritmo recursivo detallado en el apéndice A.

5.3.5. FILTRADO CEPSTRAL FRENTE A LA TRANSFORMACION BILINEAL

Las técnicas antes expuestas de modificación del conjunto de coeficientes cepstrum, tanto el filtrado cepstral como la transformación de la escala de frecuencias, han mostrado mejoras significativas en el rendimiento de sistemas de reconocimiento basados en distancia euclídea entre conjuntos de coeficientes cepstrum, no apareciendo en la bibliografía revisada resultados comparativos de dichos métodos. Sin embargo, es posible aducir razones teóricas para la elección de una técnica de filtrado cepstral frente a la transformación en la escala de frecuencias.

Como puede deducirse de las ecuaciones (5.20), los nuevos coeficientes cepstrum se obtienen como combinación lineal de los coeficientes originales, a través de la matriz de transformación W, lo que introduce correlaciones entre los nuevos coeficientes cepstrum.

Estas correlaciones aumentan con el valor de α. En la figura 5.10 se muestra la matriz de coeficientes de correlación, donde el valor de éstos se representa a través de la intensidad de gris del elemento. En la figura 5.10a se muestran la matriz de coeficientes de correlación de los cepstrum originales y en las figuras 5.10b 5.10c y 5.10d se muestra dicha matriz de correlación para valores de α=0.2, 0.4 y 0.6 respectivamente. De las figuras se desprende que la correlación es apreciable para α=0.4, y bastante alta para α=0.6, concentrándose principalmente en torno a los elementos de la diagonal.

Estas correlaciones no son deseables cuando, como se hará en las modificaciones expuestas en el siguiente capítulo, los conjuntos de coeficientes cepstrum van ha ser modelados como mezcla de gausianas, ya que la existencia de estas correlaciones obliga a considerar las matrices completas de covarianza de los coeficientes, y a no poder utilizar

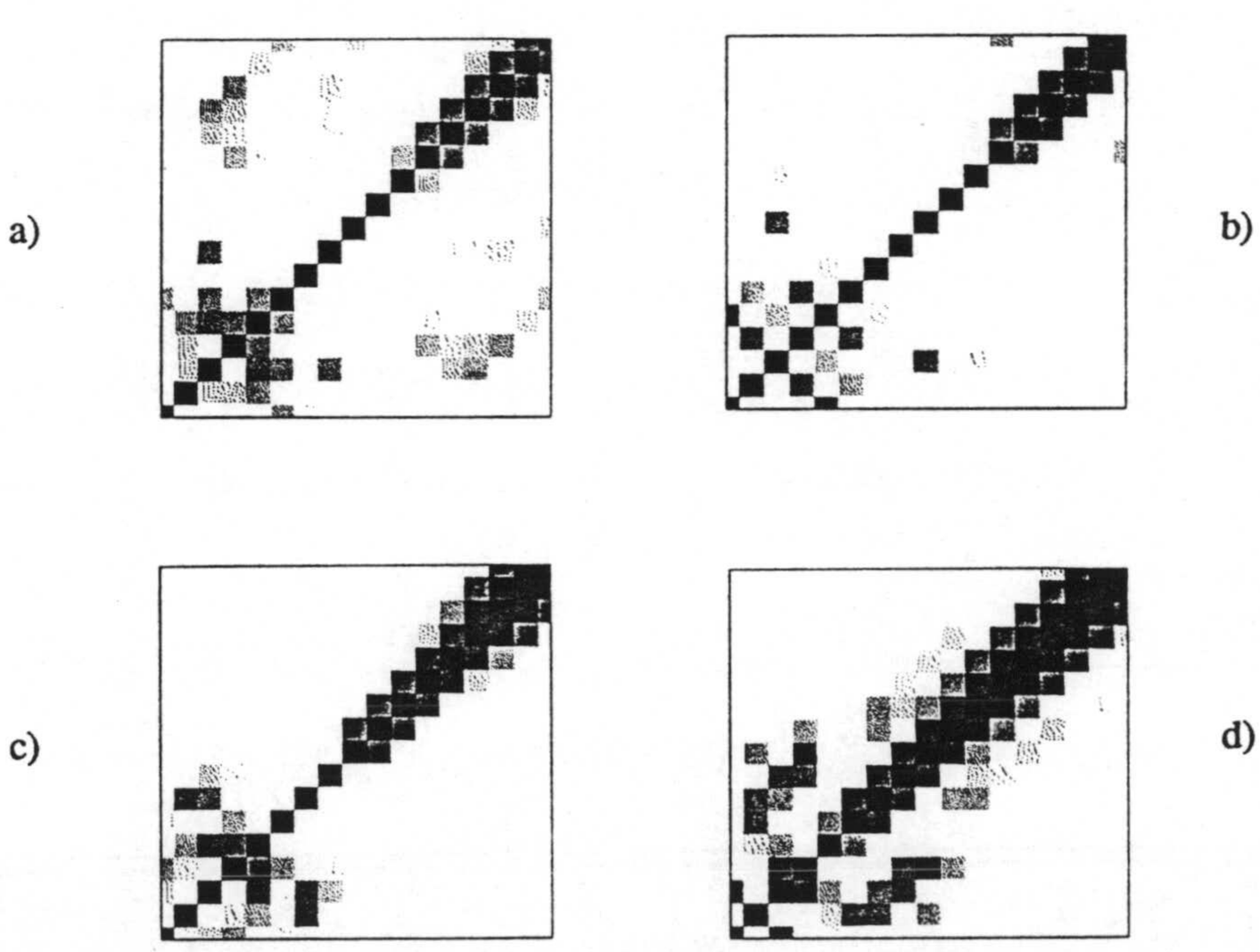

Figura 5.10. Matrices de correlación a) α=0.0 b) α=0.2 c) α=0.4 d) α=0.6

una matriz de covarianza diagonal, con los problemas de estimación que ésto conlleva, y el incremento en el coste computacional del sistema. Huang [Huang89] mostró cómo la correlación inducida por la transformación bilineal afecta al modelado con gausianas de los vectores de coeficientes cepstrum, obligando a considerar una matriz de covarianza completa, o a aumentar el número de gausianas en las mezclas. Concluyendo que la suposición de matriz de covarianza diagonal es esencialmente adecuada para vectores de coeficientes cepstrum en escala lineal, pero es inadecuada en el caso de coeficientes cepstrum transformados bilinealmente. Estas mismas consideraciones son aplicables, en cierta medida, a los coeficientes cepstrum obtenidos en escala MEL, dada la similitud entre las dos transformaciones en frecuencia.

5.3.6. SELECCION DE LA LONGITUD DE LA VENTANA CEPSTRAL

Para determinar el valor óptimo de la longitud L de la ventana de filtrado cepstral (5.11), se han realizado experimentos L1OUT multilocutor y L4OUT independiente del locutor para un sistema como el descrito en la sección 5.2.6, en el que los coeficientes cepstrum se han pesado en la forma

$$\hat{c}(n) = c(n)w(n) \qquad ; \qquad 1 \le n \le L \tag{5.21a}$$

$$w(n) = 1 + \frac{L}{2}\,\mathrm{sen}\left(\frac{n\pi}{L}\right) \qquad ; \qquad 1 \le n \le L \tag{5.21b}$$

En la tabla III y la figura 5.11 se muestran los resultados obtenidos.

LONGITUD LIFTER	MULTILOCUTOR	LOCUTOR INDEPENDIENTE
SIN LIFTER L = 10	15.07%	18.90%
L = 8	16.01%	21.78%
L = 10	13.36%	18.33%
L = 12	11.58%	16.60%
L = 14	11.53%	15.61%
L = 16	12.47%	17.33%
L = 18	12.21%	16.24%
L = 20	12.42%	15.41%

Tabla III. Error de reconocimiento en función de L.

De los resultados mostrados en la tabla III, se desprende que el valor óptimo de la longitud para la ventana de lifter es de $L = 14$, por lo que, en adelante, utilizaremos éste valor. La

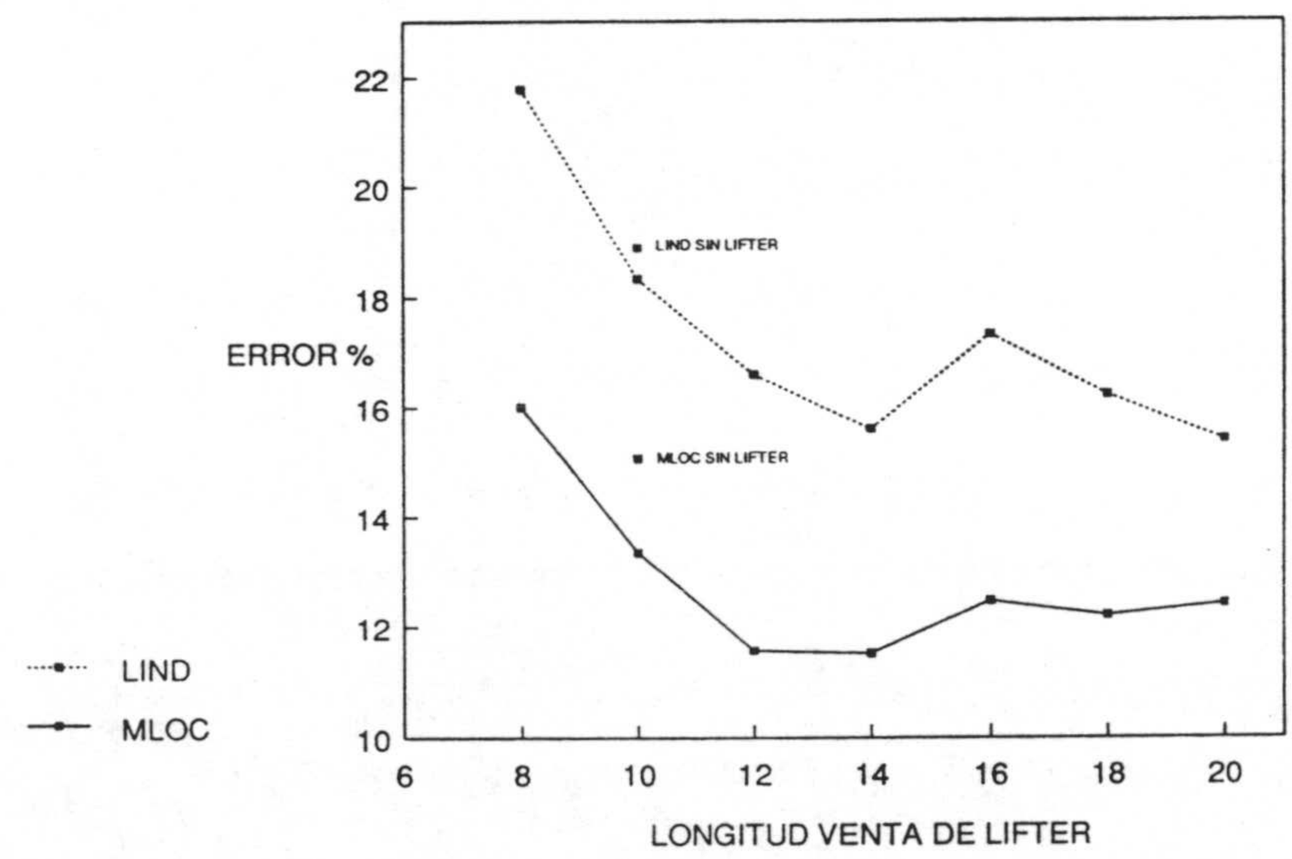

Figura 5.11. Errores de reconocimiento frente a la longitud de la ventana de *lifter*.

disminución relativa en el error de reconocimiento del sistema es de un 23.5% en el caso multilocutor y de un 17.4% en el caso independiente del locutor, mejoras que justifican la elección del pesado de los coeficientes cepstrum.

5.4. INCORPORACION DE NUEVAS CARACTERISTICAS

Las características espectrales dinámicas, correspondientes a las transiciones espectrales, son tan importantes o más que las características estáticas del espectro de la señal en la percepción humana de la voz [Ruske82], siendo las zonas de la señal donde la variación espectral es máxima, las que aportan la mayor cantidad de información fonética de las sílabas [Furui86].

Los primeros esfuerzos realizados para la incorporación de información dinámica a un sistema de reconocimiento fueron hechos por Nadas [Nadas81], definiendo una distancia entre parejas de segmentos consecutivos de señal. Este procedimiento dobla el número de parámetros a considerar, problema que fue solucionado mediante el uso de análisis de componentes principales [Duda73c]. Esta aproximación también fue utilizada por Bocchieri y Doddington [Bocchieri86]. Los principales inconvenientes de este método son el alto coste computacional, y los problemas de estimación de los elementos no diagonales de las matrices de covarianza, necesarias en el análisis de componentes principales.

De otro lado, Furui [Furui81] utilizó los primeros coeficientes de la expansión polinómica de la evolución temporal de los coeficientes cepstrum de los segmentos de de señal, mostrando que éstos parámetros son efectivos en sistemas de verificación de locutores. Esta forma de incorporar características dinámicas del espectro de la señal fue utilizada posteriormente por Furui [Furui86] en un sistema de reconocimiento de voz basado en DTW (programación dinámica), en el que dichas características se incorporaron a través de los coeficientes de regresión lineal de primer orden de los coeficientes cepstrum, mostrando que éstos son incluso más efectivos que los coeficientes cepstrum a la vista de los resultados de reconocimiento de su sistema. En el presente trabajo utilizamos el Δcepstrum tal y como se definió en la sección 5.2.2, correspondiente al primer coeficiente de regresión lineal, para incorporar información dinámica del espectro al sistema de reconocimiento.

Otro parámetro que se ha mostrado útil en el reconocimiento de voz es la energía logarítmica de la señal [Brown82]. Rabiner [Rabiner84] mostró mejoras significativas al incorporar dicho parámetro a su sistema de reconocimiento de palabras conectadas.

La energía, por si misma, no es una buena fuente de información, ya que presenta grandes variaciones entre locutores e incluso para un mismo locutor, ya que depende del volumen de locución. Es por ésto que es necesario realizar algún tipo de normalización sobre la misma. La aproximación generalmente utilizada es la de normalizar la energía de los segmentos de la señal restando el valor máximo de ésta en la palabra o frase en que están inmersos.

Por último, la Δenergía, definida en la sección 5.2.2, y que representa la evolución dinámica de la energía de la señal, también ha sido utilizado con buenos resultados. Furui [Furui86] y [Lee88b] mostraron la efectividad de este parámetro en reconocimiento de voz, incluso con resultados mejores que los obtenidos para la energía logarítmica. Además, este parámetro, al estar expresado en términos de diferencias de energías logarítmicas, no necesita ser normalizado.

5.4.1. LA INTEGRACION DE LOS PARAMETROS EN EL SISTEMA

Para la incorporación de varios parámetros a un sistema de reconocimiento basado en VQ, se han propuesto dos alternativas. La primera de ellas consiste utilizar una distancia compuesta y mantener inalterado el proceso de cuantización vectorial. La segunda alternativa consiste en suponer que los diferentes parámetros son estadísticamente independientes, y realizar un proceso de cuantización vectorial para cada uno de los diferentes tipos de parámetros, de forma que dicho proceso genere un código para cada tipo de parámetros, con lo que cada segmento de señal quedará caracterizado, no por un único símbolo, sino por un conjunto de símbolos, uno para cada tipo de parámetros considerado. A continuación describiremos estas dos aproximaciones y ofreceremos resultados comparativos de las mismas sobre nuestro sistema de reconocimiento.

Distancia compuesta

La primera aproximación a la integración de los diferentes tipos de parámetros fue propuesta por Furui [Furui86], y consiste en definir una distancia compuesta, formulada en base a una combinación lineal de las distancias entre los diferentes tipos de parámetros considerados.

Considerando un conjunto de parámetros formado por el cepstrum, Δcepstrum, energía y Δenergía en la forma

$$x = (\{c(n)\}_{n=1..L}, \{\Delta c(n)\}_{n=1..L}, E, \Delta E) \tag{5.22}$$

definiremos una distancia compuesta en la forma siguiente

$$d(x_r, x_s) = w_c\, d_c(c_r, c_s) + w_{dc}\, d_{dc}(\Delta c_r, \Delta c_s)$$
$$+ w_e\, d_e(E_r, E_s) + w_{de}\, d_{de}(\Delta E_r, \Delta E_s) \tag{5.23a}$$

$$d_c(c_r, c_s) = \sum_{n=1}^{L} [\, c_r(n) - c_s(n)\,]^2 \tag{5.23b}$$

$$d_{dc}(\Delta c_r, \Delta c_s) = \sum_{n=1}^{L} [\, \Delta c_r(n) - \Delta c_s(n)\,]^2 \tag{5.23c}$$

$$d_e(E_r, E_s) = [\, E_r - E_s\,]^2 \tag{5.23d}$$

$$d_{de}(\Delta E_r, \Delta E_s) = [\, \Delta E_r - \Delta E_s\,]^2 \tag{5.23d}$$

donde los pesos w_c, w_{dc}, w_e y w_{de} se eligen experimentalmente para rendimiento óptimo del sistema en términos del error de reconocimiento.

Con esta definición de distancia, y una vez seleccionados los valores para los pesos, el sistema de reconocimiento no se ve modificado más que en el proceso de cuantización vectorial debido a la nueva definición de la función distancia.

Múltiples observaciones

Una aproximación alternativa es la propuesta por Gupta [Gupta87] y utilizada por Lee [Lee88b], que se basa en la suposición de independencia estadística entre los diferentes tipos de parámetros. La cuantización vectorial se realiza diseñando un diccionario para cada parámetro y generando un conjunto de símbolos, en lugar de uno único, para cada segmento de la señal. De esta forma, las probabilidades de observación de los estados del modelo son producto de las probabilidades de observación para cada uno de los diferentes tipos de símbolos, tal y como se indica en 5.24.

$$b_i(o_t) = \prod_{j=1}^{c} b_{ij}(o_t{}^j) \tag{5.24a}$$

$$y_t = (o_t{}^1, o_t{}^2, \cdots, o_t{}^c) \tag{5.24b}$$

donde o_t^j denota el símbolo observado del tipo j en el instante de tiempo t, y c es el número de símbolos por observación. Las fórmulas de evaluación y reestimación han de modificarse consecuentemente para tener en cuenta la nueva definición de probabilidades de observación para los modelos.

A continuación describiremos los resultados obtenidos para las configuraciones multilocutor y independiente del locutor del sistema utilizando las dos aproximaciones antes citadas, comenzando por la correspondiente a la integración a través de una distancia compuesta.

5.4.2. DISTANCIA COMPUESTA

A continuación mostraremos las modificaciones necesarias en el sistema para la incorporación de las nuevas características en el caso de utilización de una distancia compuesta. En primer lugar describiremos el proceso de elección de los pesos óptimos para la composición de distancias, y a continuación discutiremos la determinación de los nuevos valores óptimos para el número de estados de los modelos y el número de símbolos en el diccionario de cuantización vectorial. Por último describiremos los resultados obtenidos para la nueva configuración del sistema.

Para la determinación de los pesos óptimos en la composición de distancias, reformularemos los valores de éstos en términos de las varianzas totales de cada uno de los tipos de parámetros en la forma siguiente

$$d(x_r, x_s) = \frac{\tilde{w}_c}{V_c} d_c(c_r, c_s) + \frac{\tilde{w}_{dc}}{V_{dc}} d_{dc}(\Delta c_r, \Delta c_s)$$

$$+ \frac{\tilde{w}_e}{V_e} d_e(E_r, E_s) + \frac{\tilde{w}_{de}}{V_{de}} d_{de}(\Delta E_r, \Delta E_s) \qquad (5.25a)$$

$$\tilde{w}_c = V_c w_c \qquad V_c = \sum_{n=1}^{L} \sigma^2_{c_n}$$

$$\tilde{w}_{dc} = V_{dc} w_{dc} \qquad V_{dc} = \sum_{n=1}^{L} \sigma^2_{dc_n} \qquad (5.25b)$$

$$\tilde{w}_e = V_e w_e \qquad V_e = \sigma^2_e$$

$$\tilde{w}_{de} = V_{de} w_{de} \qquad V_{de} = \sigma^2_{de}$$

Donde V_c, V_{dc}, V_e y V_{de} representan las varianzas totales de cada uno de las diferentes características, y σ^2 representa la varianza. Esta normalización se realiza para que los pesos $\tilde{w}$ sean del mismo orden de magnitud.

Determinación del peso del Δcepstrum

En primer lugar, determinaremos el peso relativo del Δcepstrum, dado que esta característica es más eficiente que las correspondientes a la energía y Δenergía. Para hacer ésto utilizaremos una distancia con $\tilde{w}_e = \tilde{w}_{de} = 0$ en la forma

$$d(x_r, x_s) = \tilde{w}_c \left[\frac{d_c(c_r, c_s)}{V_c}\right] + \tilde{w}_{dc} \left[\frac{d_{dc}(\Delta c_r, \Delta c_s)}{V_{dc}}\right] \qquad (5.26a)$$

$$= (1 - \mu_{dc}) \left[\frac{d_c(c_r, c_s)}{V_c}\right] + \mu_{dc} \left[\frac{d_{dc}(\Delta c_r, \Delta c_s)}{V_{dc}}\right] \qquad (5.26b)$$

en la que, haciendo variar μ_{dc} entre 0 y 1 se consigue cubrir el rango de posibles valores para la combinación de las dos características. Los pesos óptimos están relacionados con μ_{dc} según

$$\frac{\tilde{w}_{dc}}{\tilde{w}_c} = \frac{\mu_{dc}}{(1-\mu_{dc})} \tag{5.27}$$

de forma relativa, ya que un factor global de escala no afecta al rendimiento de la distancia en el sistema.

Para la determinación del valor óptimo de μ_{dc}, se ha realizado un experimento en el que dos secuencias de cada palabra para cada uno de los locutores de la base de datos se han utilizado como conjunto de entrenamiento, y la tercera como conjunto de test. Los resultados se muestran en la figura 5.12. En la figura 5.12a se muestra la variación del error de reconocimiento del sistema y en la figura 5.12b la variación de la distorsión media del diccionario. El valor óptimo de μ_{dc} está en el rango 0.4 a 0.7, obteniéndose el mínimo error para el valor $\mu_{dc} = 0.65$, que corresponde con el máximo en la distorsión media del diccionario, que a su vez coincide con el valor para el que la composición alcanza el valor máximo de varianza estadística.

Estos valores corresponden a unos pesos $\tilde{w}_c = 1$ y $\tilde{w}_{dc} = 1.857$, para los que se obtiene un error de 6.88% frente al 11.25% obtenido utilizando únicamente el cepstrum

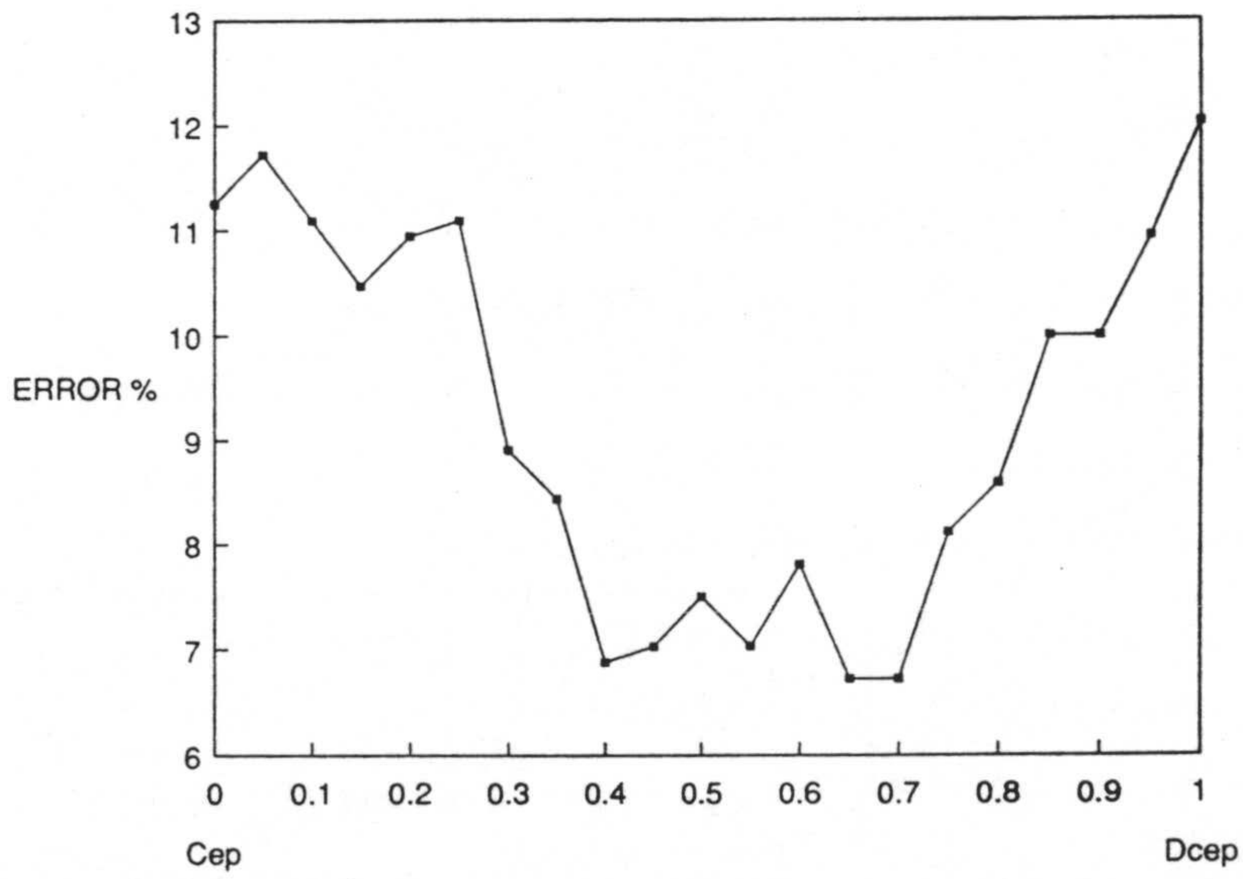

Figura 5.12a. Error frente a la combinación Cep/Dcep

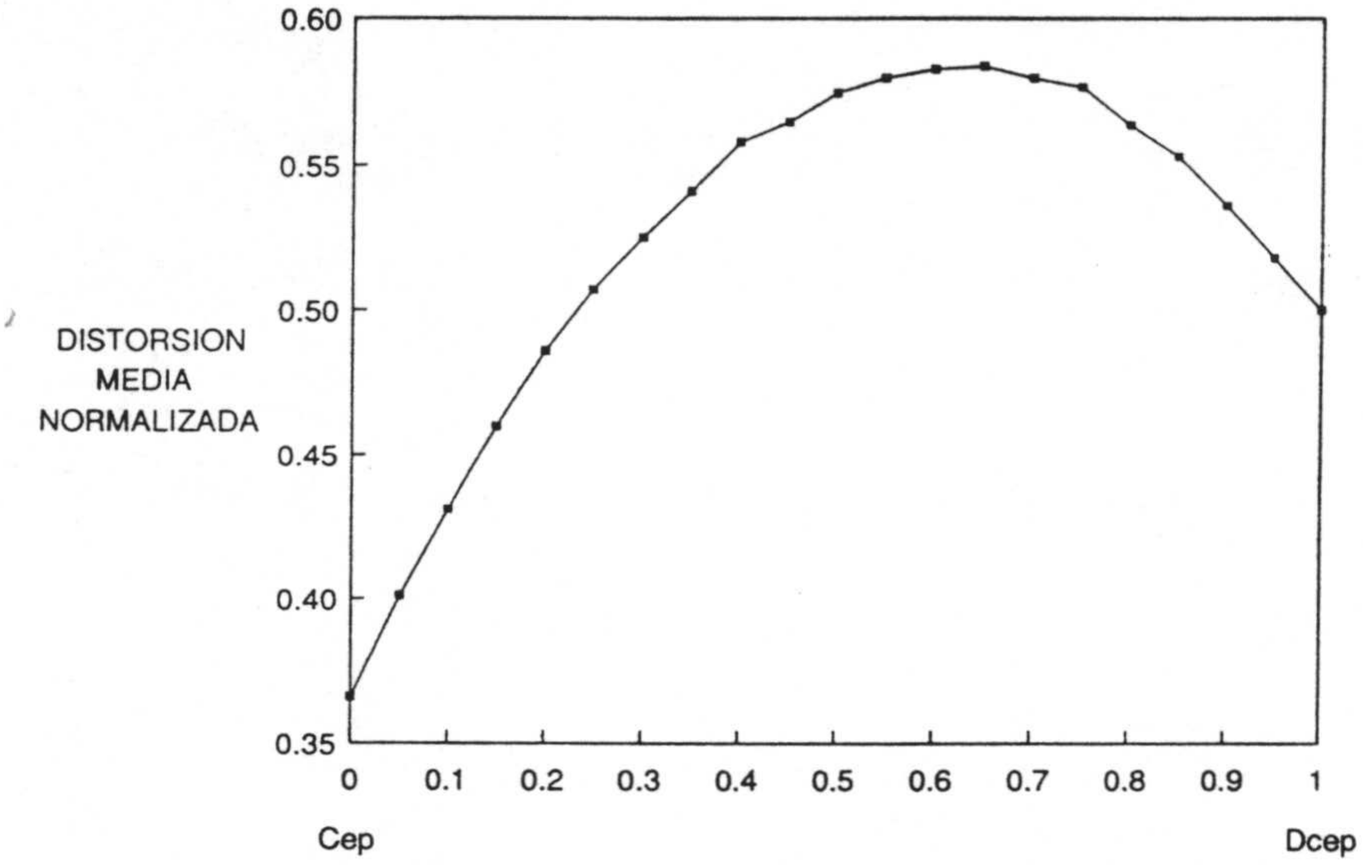

Figura 5.12b. Distorsión media frente a la combinación Cep/Dcep

($\mu_{dc} = 0$) y al 12.03% obtenido utilizando sólo el Δcepstrum ($\mu_{dc} = 1$), lo que indica que la composición de las dos características produce un rendimiento mayor que el obtenido para cada una de ellas por separado, y justifica la utilización conjunta de las dos características.

Determinación de los pesos para la energía y la Δenergía

Una vez fijado el peso del Δcepstrum, y dado que en experimentos previos se determinó que la Δenergía es una característica más efectiva que la energía, se procedió a la sintonización del peso de la Δenergía, para lo que se formuló una distancia en la forma

$$d(x_r, x_s) = (1-\mu_{de})\left\{\tilde{w}_c \left[\frac{d_c(c_r, c_s)}{V_c}\right] + \tilde{w}_{dc}\left[\frac{d_{dc}(\Delta c_r, \Delta c_s)}{V_{dc}}\right]\right\} + \mu_{de}\left[\frac{d_{de}(\Delta E_r, \Delta E_s)}{V_{de}}\right] \tag{5.28}$$

en la que los valores $\tilde{w}_c$ y $\tilde{w}_{dc}$ se fijaron a 1.000 y 1.857 respectivamente, y se varió μ_{de} en el intervalo [0,1] en un experimento como el descrito en el apartado anterior con los

resultados que se muestran el la figura 5.12. El valor seleccionado para el peso relativo de la Δenergía es el correspondiente a $\mu_{de} = 0.35$. con un error de 5.00% frente al 6.88% obtenido en la configuración anterior. El valor correspondiente de $\tilde{w}_{de}$ es

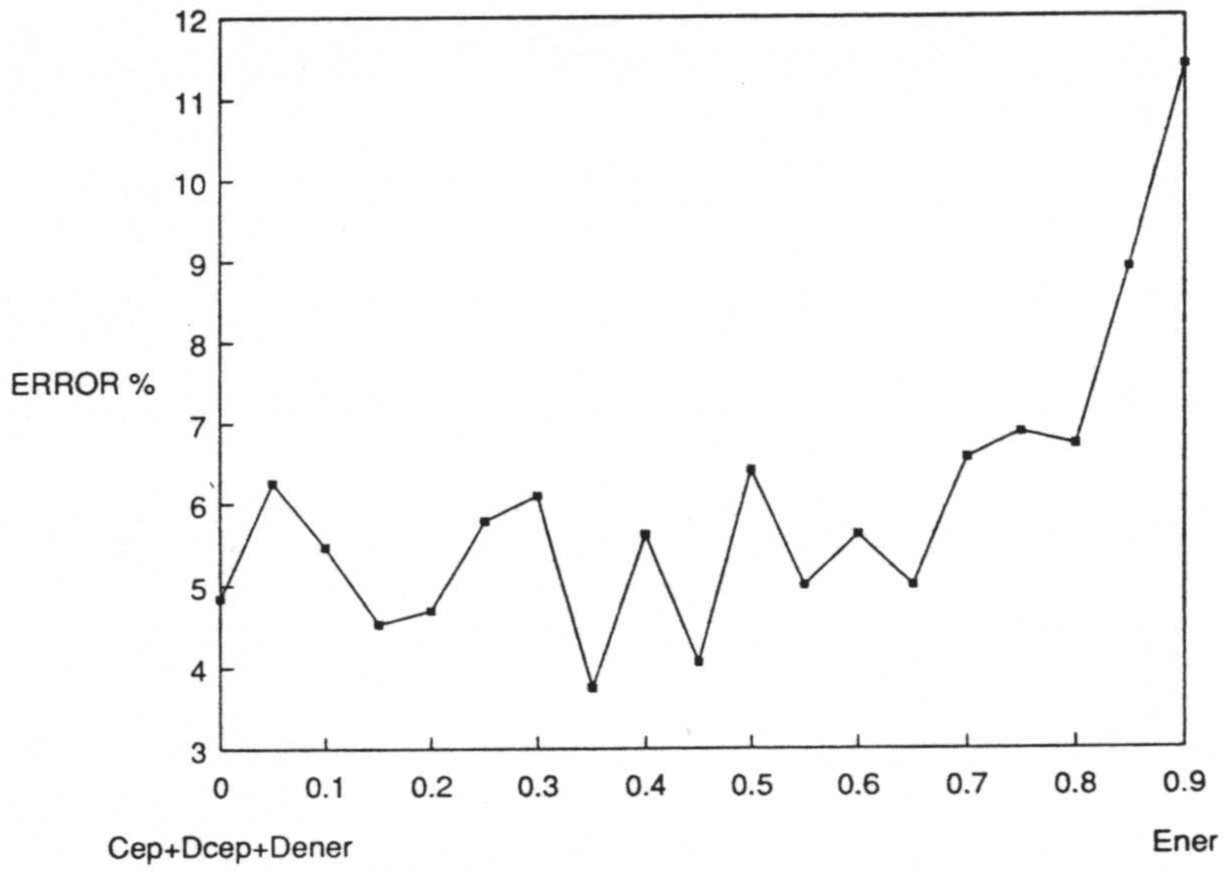

Figura 5.13. Sintonización del peso de la Δenergía

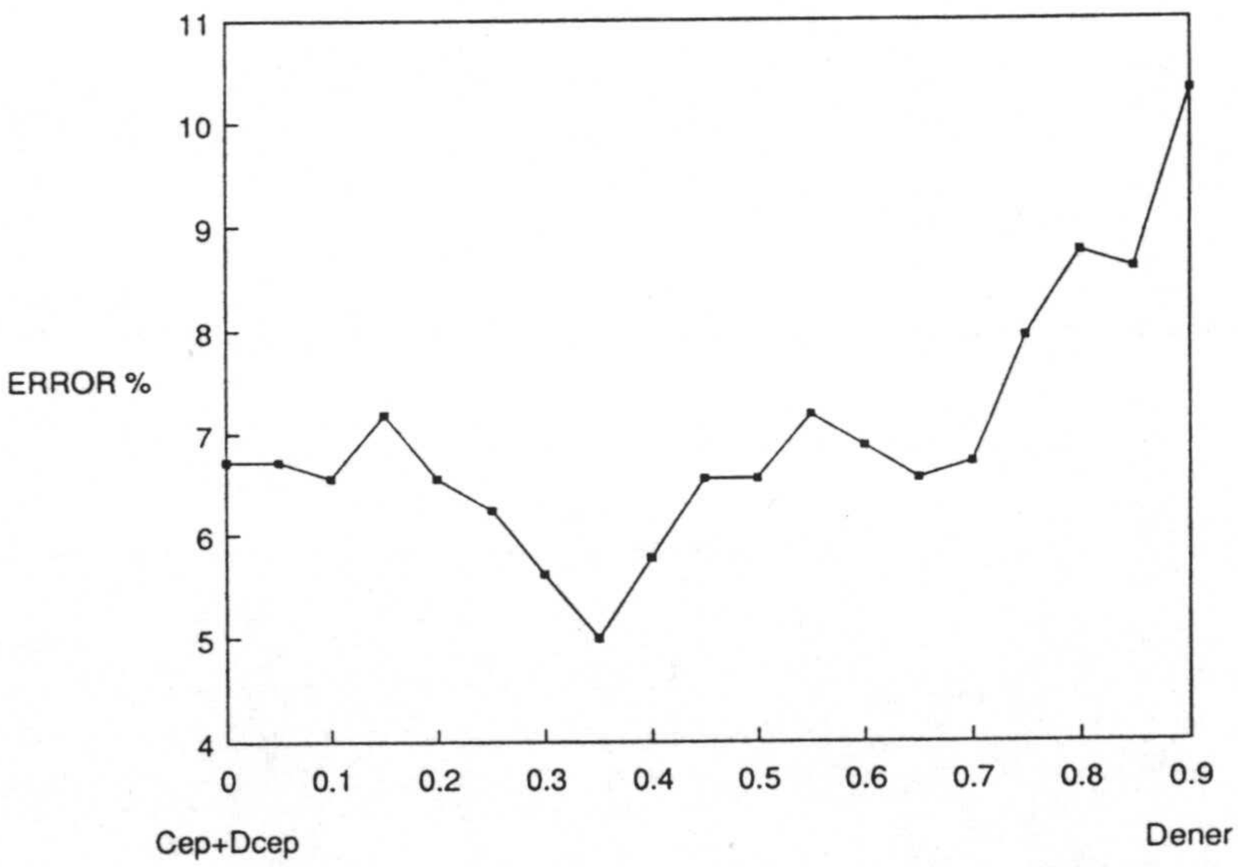

Figura 5.14. Sintonización del peso de la energía

$$\tilde{w}_{de} = \frac{\mu_{de}}{1 - \mu_{de}} = 0.538 \tag{5.29}$$

Por último, se procedió, de forma análoga a los casos anteriores, para la determinación del valor óptimo para la composición de la energía. Los resultados obtenidos se muestran en la figura 5.14. Como puede apreciarse en la figura, no existe un mínimo absoluto bien definido para el error en función de la adición de la energía a la función distancia, sino que el error oscila en torno al valor medio obtenido sin este parámetro, por lo que se decidió no incorporar dicho parámetro a la función distancia.

En base a los resultados obtenidos en éste y el experimento previo, los pesos seleccionados para la distancia son los indicados en la tabla IV. Estos valores son los utilizados en el resto del trabajo.

$\tilde{w}_c$	$= 1.000$
$\tilde{w}_{dc}$	$= 1.857$
$\tilde{w}_e$	$= 0.000$
$\tilde{w}_{de}$	$= 0.538$

Tabla IV. Pesos óptimos para la distancia compuesta.

5.4.3. RESINTONIZACION DE LOS PARAMETROS DEL SISTEMA

Una vez modificada la distancia para incorporar las nuevas características al sistema, hay dos parámetros del mismo, el número de centros del diccionario VQ y el número de estados de los modelos HMM, que necesitan ser determinados para esta nueva configuración. El motivo es que al incrementar el número de parámetros del vector representativo de cada segmento de señal, también se aumenta la varianza total del vector de características, y por lo tanto aumenta la distorsión del diccionario, por lo que es necesario aumentar el número de centros de éste para modelar adecuadamente el nuevo conjunto de vectores de caracterísiticas.

De otro lado, la inclusión de características dinámicas del espectro de la señal, hace que sea posible describir más detalladamente la evolución temporal de las características espectrales, asignando estados de los modelos no sólo a las zonas estacionarias del espectrograma de la señal, sino también a las transicionales, lo que se consigue añadiendo nuevos estados a los modelos HMM.

Para la determinación del valor ótimo de estos dos parámetros del sistema, se realizaron experimentos L1OUT para la configuración multilocutor y L4OUT para la configuración independiente del locutor. En la figura 5.15 se muestran los resultados obtenidos para la variación del error con el número de estados Ne de los modelos HMM y el número de centros del diccionario Ns. La elección Ne=10 corresponde, en la mayoría de las situaciones, a un valor cercano al óptimo, variando poco para diferentes tamaños de diccionario, por lo que en adelante, este será el valor seleccionado para el número de estados de los modelos HMM. Nótese que este valor corresponde aproximadamente al doble del número medio de fonemas en las palabras de la base de datos, aproximadamente igual al número de zonas estables y transicionales de las palabras. En la tabla V se muestra la variación del error con el número de centros del diccionario VQ para un número de estados Ne = 10.

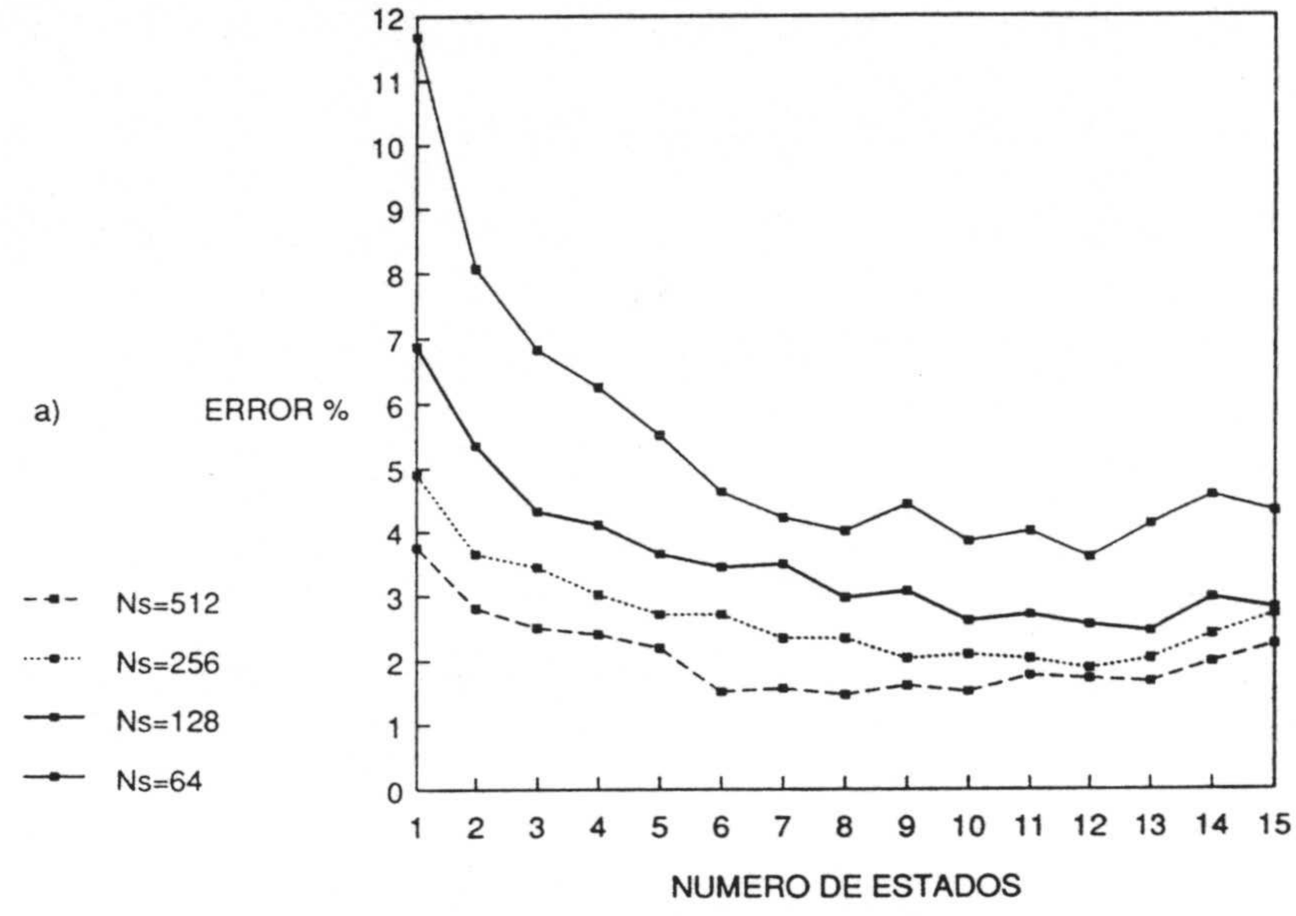

Figura 5.15a. Error frente a Ne y Ns. Experimento L1OUT

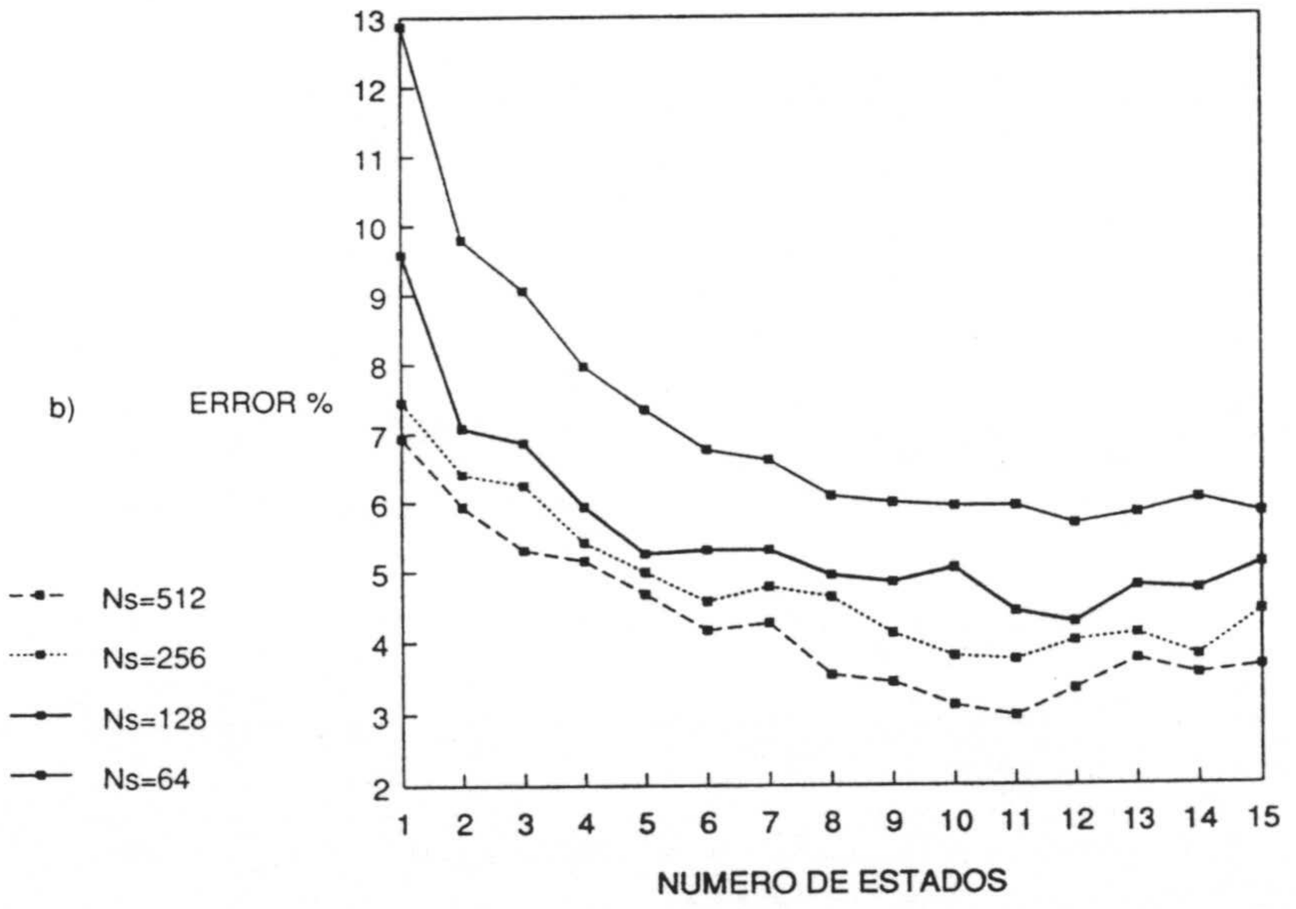

Figura 5.15b. Error frente a Ne y Ns. Experimento L4OUT

Ns	L1OUT	L4OUT
64	3.85%	5.21%
128	2.61%	4.48%
256	2.09%	3.44%
512	1.51%	3.33%

Tabla V. Error de reconocimiento en función de Ns para Ne=10.

5.4.4. MODELOS DE MARKOV VECTORIALES

Con esta denominación haremos referencia, en adelante, a los modelos discretos de Markov con múltiples símbolos observables, aludiendo al hecho de que en lugar de caracterizar cada segmento de señal con un único símbolo, los caracterizaremos por un vector de símbolos correspondientes a diferentes características observables, que denominaremos vector de símbolos observables.

En la implementación utilizada de este tipo de modelos, utilizamos un diccionario para vectores cepstrum, otro para vectores Δcepstrum, y un tercero para vectores energía-Δenergía. Los vectores cepstrum y Δcepstrum se pesaron con una ventana de *liftering* seno-remontado de longitud L=14, y se utilizó una distancia euclídea en la construcción del diccionario. En cuanto al diccionario de energías, se utilizaron vectores de dos componentes formados por parejas de valores formadas por la energía y la Δenergía con una distancia euclídea pesada en la forma

$$d_{e-de} = \frac{[E_r - E_s]^2}{\sigma^2_E} + \frac{[\Delta E_r - \Delta E_s]^2}{\sigma^2_{\Delta E}} \tag{5.30}$$

donde σ^2_E y $\sigma^2_{\Delta E}$ representan las varianzas de la energía y Δenergía respectivamente, estimadas sobre los vectores del conjunto de entrenamiento, con lo que se ecualizan las contribuciones de los dos parámetros a la distancia total.

Los parámetros de los modelos HMM se fijaron igual que en los experimentos anteriores, con 10 estados para todos los modelos. Los resultados obtenidos para los

experimentos L1OUT y L4OUT se muestran en la tabla VI. Los resultados de esta tabla corresponden a tres configuraciones para los modelos vectoriales; en la primera de éstas todos los diccionarios se construyeron utilizando el mismo número de centros, lo que corresponde a una configuración VQ con el mismo coste computacional que un diccionario único con distancia compuesta del mismo número de centros. La segunda configuración corresponde a la elección de un número de centros mitad para el diccionario de energías, similar a la utilizada por Mariño [Mariño90], en esta situación el coste computacional del proceso VQ es similar aunque algo menor que en la situación anterior. Por último, también se han realizado pruebas sobre una configuración con un número de centros para el diccionario de energías igual a la cuarta parte del número de centros de los otros dos diccionarios.

N_C–$N_{\Delta C}$–$N_{E-\Delta E}$	N_{TOT}	L1OUT	L4OUT
64-64-64	192	2.81%	4.06%
64-64-32	160	2.40%	3.74%
64-64-16	144	3.07%	3.90%
128-128-128	384	2.34%	3.17%
128-128-64	320	2.03%	3.12%
128-128-32	288	2.34%	3.44%
256-256-256	768	2.40%	3.91%
256-256-128	640	1.82%	3.12%
256-256-64	576	1.72%	3.07%

Tabla VI. Error de reconocimiento para los modelos vectoriales.

5.4.5. MULTIPLES OBSERVACIONES FRENTE A DISTANCIA COMPUESTA

De la comparación de los resultados de reconocimiento obtenidos para los modelos vectoriales y los modelos standard con distancia compuesta, mostrados en las tablas V y VI, para un número de centros igual o similar, se desprende que las diferencias en las tasas de error son pequeñas, del orden de 0.37% en los experimentos L1OUT y L4OUT para 256+256+64 centros en los modelos vectoriales frente a 256 centros en los modelos con

distancia compuesta. Para un número de centros menor (64 y 128), las diferencias son mayores, del orden de 0.58-1.36% para 128 centros y de 1.45-1.74% para 64 centros.

Estos resultados indican que la utilización de modelos vectoriales sólo reporta mejoras significativas cuando el número de centros considerado es reducido (p.e. 64), pero estas mejoras son menores al aumentar el número total de centros (p.e. 128 y 256). Además, la utilización de modelos vectoriales implica aumentar los requerimientos de memoria de los modelos (que aumentan aproximadamente en un factor 3), y la complejidad introducida en el cálculo de las probabilidades de generación de tripletes de símbolos.

A la vista de estos resultados, y teniendo en cuenta la mayor complejidad y coste computacional de los modelos vectoriales, decidimos utilizar modelos discretos con distancia compuesta en el resto del trabajo.

5.5. DURACION DE ESTADOS

Como ya indicamos en la sección 4.5, ecuación (4.16), la probabilidad de obtener d observaciones consecutivas en un mismo estado s_i tiene una distribución exponencial de la forma

$$P_i(d) = (a_{ii})^{d-1}(1-a_{ii})$$

Para muchos procesos físicos, y en concreto para modelos de Markov de señales de voz, la distribución de probabilidad de duración de estados no responde a esta distribución.

Para solucionar este problema se han propuesto varias aproximaciones diferentes, la primera consiste en asumir una cierta distribución de probabilidad paramétrica, y modificar las fórmulas de evaluación y reestimación consecuentemente con esta definición [Rabiner89a, Chang90]. Estas aproximaciones conllevan un incremento significativo en el coste computacional del sistema, por lo que se han propuesto aproximaciones alternativas, como la limitación de las duraciones mínima y máxima de los estados del modelo [Gu90] o la incorporación de información sobre la duración de los estados de los modelos como un postprocesador en el sistema [Rabiner89].

En esta última aproximación, se realiza una estimación no paramétrica de las distribuciones de probabilidad $P_i(d)$ para los diferentes estados de los modelos simplemente a través de la decodificación en estados de las secuencias del conjunto de entrenamiento mediante el algoritmo de Viterbi. Una vez estimadas estas distribuciones de probabilidad, y dada la decodificación en estados de una secuencia, podemos evaluar

$$P_{DUR}(X_1^T|\lambda) = \sum_{i=1}^{Ne} \log[P_i(d_i)] \tag{5.31a}$$

donde d_i representa la duración del estado s_i de la secuencia, y combinarla con la probabilidad de generación logarítmica

$$P_{GEN}(X_1^T|\lambda) = \log[P(X_1^T|\lambda)] \tag{5.31b}$$

obteniendo una probabilidad total en la forma

$$P_{TOT}(X_1^T|\lambda) = P_{GEN}(X_1^T|\lambda) + w_{DUR}\, P_{DUR}(X_1^T|\lambda) \tag{5.31c}$$

Este método de incorporar la duración de los estados a la probabilidad logarítmica de generación genera resultados esencialmente iguales a los obtenidos mediante un modelado explícito de la duración de los estados [Rabiner89], por lo que, en el presente trabajo utlizaremos esta técnica de incorporación de la información duracional al modelo.

5.5.1. SINTONIZACION DEL PESO PARA LA DURACION DE LOS ESTADOS

Para la determinación del valor óptimo de w_{DUR}, se realizaron experimentos L1OUT y L4OUT variando el parámetro w_{DUR} en el intervalor [0,1], para diferente número de centros en el diccionario, obteniendo los resultados que se muestran en las curvas de la figura 5.16. El valor óptimo para w_{DUR} depende del número de centros del diccionario, pero el valor 0.5 está próximo al punto óptimo para la mayoría de los casos, y es el valor que se elige para la composición de las dos probabilidades logarítmicas. Los resultados se muestran en la tabla VI.

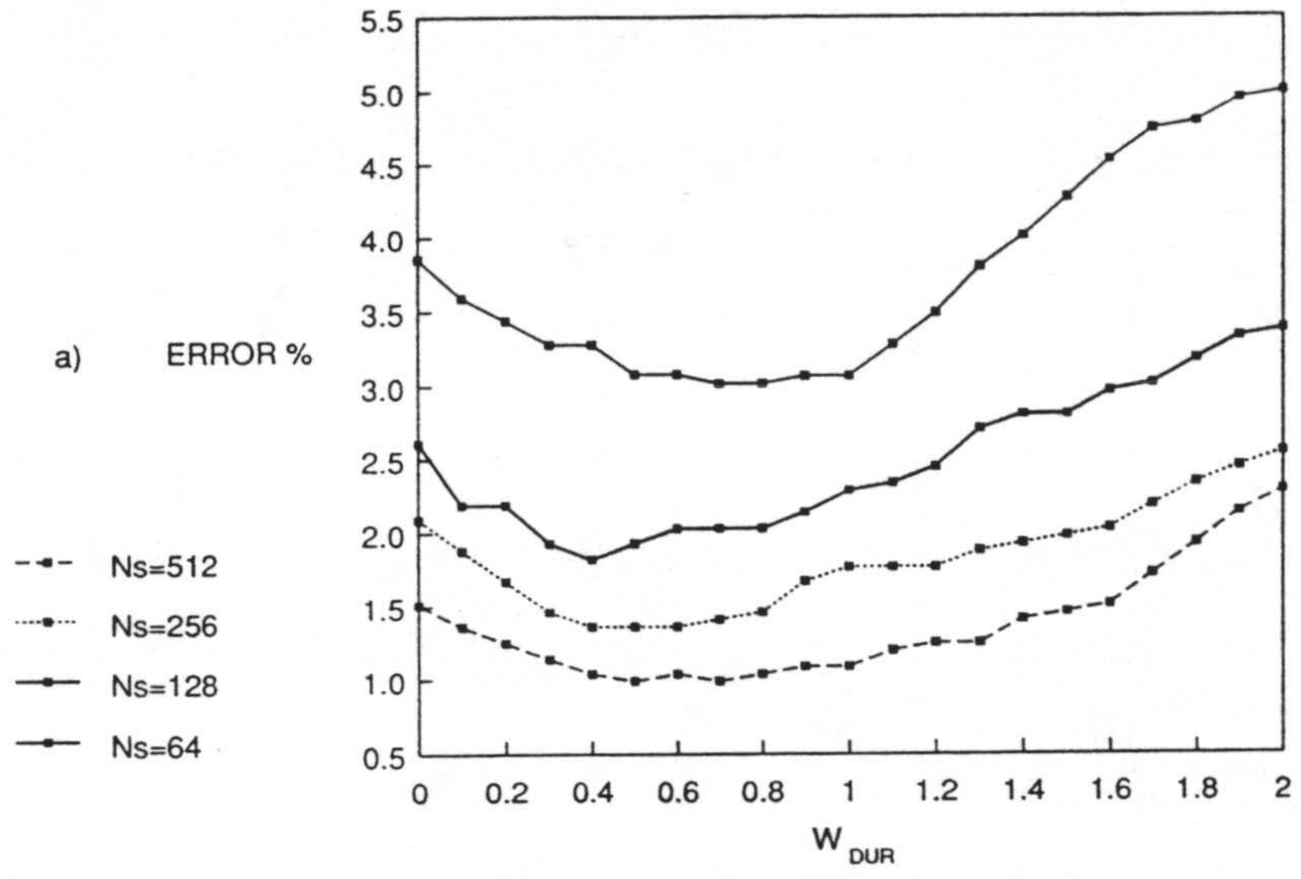

Figura 5.16a. Error frente a la composición Pgen/Pdur. Experimento L1OUT

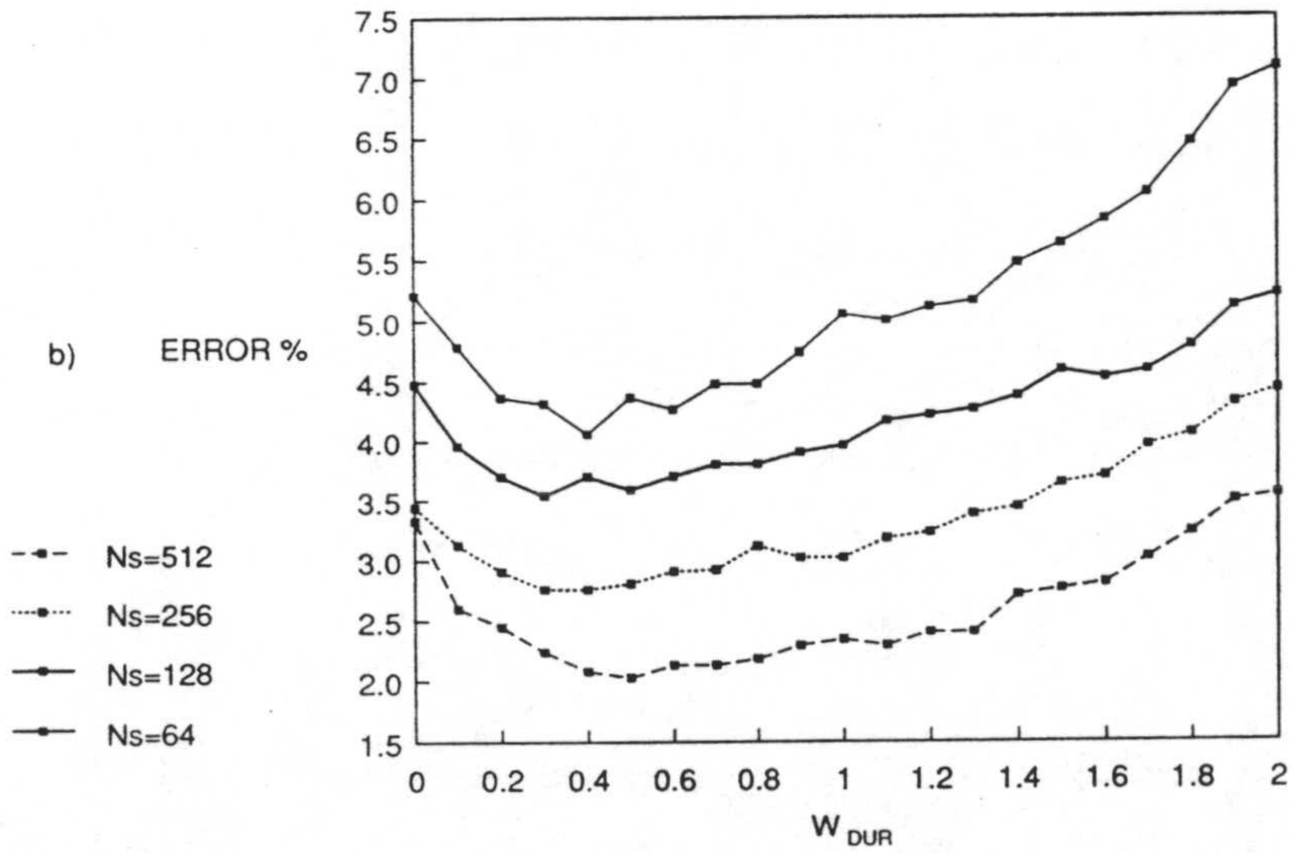

Figura 5.16b. Error frente a la composición Pgen/Pdur. Experimento L4OUT

Ns	L1OUT	L4OUT
64	3.08%	4.37%
128	1.93%	3.59%
256	1.36%	2.81%
512	0.99%	2.03%

Tabla VI. Error de reconocimiento con duración de estados.

5.6. RESUMEN DE RESULTADOS PARA EL SISTEMA DE REFERENCIA

Para referencia del siguiente capítulo, resumimos los resultados finales obtenidos para el sistema básico de reconocimiento. En la tabla VII se muestran los resultados obtenidos para cada una de las modificaciones introducidas al sistema. En la columna TIPO, la entrada RECT indica lifter rectangular (sin pesado en la distancia), y la entrada SIN representa una ventana de liftering del tipo seno-remontado. La columna L representa la longitud de la ventana de lifter (número de coeficientes) y la columna DUR representa la inclusión o no de la duración de estados en la evaluación del sistema. Las columnas *Ne* y *Ns* indican el número de estados de los modelos y de símbolos del diccionario VQ respectivamente.

CONFIGURACION						ERROR	
PARAMETROS	Ne	Ns	LIFTER		DUR	L1OUT	L4OUT
			TIPO	L			
Cepstrum	5	64	RECT	10	NO	15.07%	18.90%
Cepstrum	5	64	SIN	14	NO	11.53%	15.61%
Distancia	10	64	SIN	14	NO	3.85%	5.21%
compuesta	10	128	SIN	14	NO	2.61%	4.48%
C+ΔC+ΔE	10	256	SIN	14	NO	2.09%	3.44%
	10	512	SIN	14	NO	1.51%	3.33%
Distancia	10	64	SIN	14	SI	3.08%	4.37%
compuesta	10	128	SIN	14	SI	1.93%	3.59%
C+ΔC+ΔE	10	256	SIN	14	SI	1.36%	2.81%
	10	512	SIN	14	SI	0.99%	2.03%

Tabla VII. Resumen de resultados para el sistema de referencia

CAPITULO 6

MEJORAS AL SISTEMA DE RECONOCIMIENTO

6.1. INTRODUCCION

En los capítulos anteriores hemos mostrado el desarrollo de un sistema de reconocimiento de palabras aisladas basado en modelos discretos de Markov y cuantización vectorial (DHMM).

Se ha mostrado como mejorar el rendimiento de dicho sistema actuando sobre la representación paramétrica de la señal, tanto a través de una modificación de los parámetros considerados (liftering), como a través de la incorporación de nuevos parámetros en representación de la información dinámica del espectro de la señal.

Por último, se ha mostrado cómo modificar la probabilidad de generación de los modelos de Markov incorporando información sobre la duración de los estados obtenidos a través del algoritmo de decodificación de Viterbi, minimizando de esta forma los efectos del modelado incorrecta que de dichas duraciones realizan los modelos de Markov.

En este capítulo mostraremos como se puede incrementar el rendimiento del sistema de reconocimiento modificando el proceso de cuantización vectorial del mismo. Presentaremos dos posibles modificaciones, los primera de éstas intenta minimizar los errores introducidos en el proceso de cuantización. Para ésto introduciremos el concepto de Modelos Semicontínuos de Markov (SCHMM) [Huang89b, Huang89a, Huang89c], e indicaremos cómo modificar el sistema de reconocimiento para utilizar éste tipo de modelado.

La segunda modificación está motivada por los resultados obtenidos en trabajos previos en los que se mostró cómo las distorsiones de cuantización obtenidas en el proceso

VQ pueden utilizarse como información útil en el reconocimiento de palabras aisladas [Burton85b, Pan85a, Furui88]. Básicamente, el proceso consiste en diseñar un diccionario VQ para cada una de las palabras del vocabulario, y utilizar la distorsión de cuantización de cada palabra incógnita con los diferentes diccionarios VQ en el proceso de decisión del sistema.

En este capítulo mostraremos cómo utilizar esta información en el contexto del modelado HMM discreto, desarrollaremos una nueva variante de dicho modelado discreto que denominaremos modelos de Markov con cuantización dependiente MVQHMM aludiendo al hecho de que en este tipo de modelos discretos, el proceso de cuantización se realiza de forma independiente para cada uno de los modelos correspondientes a las palabras del vocabulario, a través de un diccionario VQ específico.

El resto del capítulo está estructurado en la forma siguiente. En la sección 6.2 se introducirá el concepto de modelado semicontínuo de Markov y se indicarán las modificaciones necesarias en las fórmulas de evaluación y reestimación utilizadas en el modelado discreto de Markov para adecuarlas al nuevo tipo de modelado. Por último se describirá la implementación utilizada en este trabajo.

En la sección 6.3 se mostrarán los resultados obtenidos en experimentos de sintonización de parámetros para los modelos semicontínuos y los resultados de reconocimiento para configuraciones multilocutor e independiente del locutor de la implementación del sistema con modelos semicontínuos.

En la sección 6.4 se introducirá intuitivamente el concepto de modelo de Markov con cuantización dependiente, resumiendo en primer lugar los resultados obtenidos en los trabajos que han motivado el desarollo de esta variante del modelado HMM discreto.

En la sección 6.5 se desarrolla formalmente el modelado MVQHMM, y se derivan las relaciones necesarias para la obtención de las probabilidades de generación de este tipo de modelos de Markov. Por último, se describe el proceso de construcción de los modelos MVQHMM, así como la forma en que éstos son utilizados para la clasificación de palabras.

En la sección 6.6 se describe la implementación utilizada en este trabajo para el modelado MVQHMM, y se presentan y discuten los resultados obtenidos para

experimentos multilocutor y independiente del locutor.

En la sección 6.7 se comparan los resultados obtenidos con los tres tipos de modelado presentados en esta memoria, tanto desde el punto de vista del rendimiento obtenido en reconocimiento, como desde el punto de vista de su complejidad computacional.

Por último, en la sección 6.8 se discute el problema de la reducción del número total de centros utilizado en el sistema de reconocimiento basado en modelos MVQHMM, se describe un algoritmo para efectuar dicha reducción y se presentan los resultados obtenidos.

6.2. MODELOS SEMICONTINUOS DE MARKOV

El proceso de discretización inherente al modelado discreto de Markov permite realizar una estimación no paramétrica de las probabilidades de producción de observaciones de los estados del modelo. De esta forma, se realiza una formulación general de dichas probabilidades. Sin embargo, este proceso de discretización introduce errores de cuantización que provocan una pérdida de información que a su vez deteriora el rendimiento del sistema [Rabiner85]. Estos errores se producen en situaciones típicas en las que un vector se encuentra entre dos centros (geométricamente hablando) del diccionario VQ.

De otro lado, el modelado HMM contínuo obvia este problema al realizar un modelado paramétrico de las funciones densidad de probabilidad contínuas de producción de los modelos. Sin embargo, este tipo de modelado tiene el inconveniente de que es necesario elegir una determinada forma paramétrica para la funciones densidad de probabilidad, con la pérdida de generalidad que ésto conlleva.

Para mantener la generalidad en la formulación de los modelos contínuos, se utiliza normalmente una combinación lineal de gausianas multivariadas en la forma que se indicó en la sección 3.4. Trabajos previos [Rabiner85] mostraron que el modelado contínuo de Markov mediante mezcla de gausianas multivariadas es adecuado para señales de voz, caracterizadas por diferentes tipos de parámetros, incluso bajo la suposición de matrices de covarianza diagonal, considerando un número suficientemente elevado de componentes en la combinación lineal.

El principal inconveniente de este tipo de modelado es el elevado número de parámetros que es necesario estimar. Como ejemplo consideremos un modelo contínuo cuyas densidades de probabilidad de observación son modeladas con una mezcla de M=7 gausianas con matrices de covarianza diagonales, y que el vector de parámetros está formado por D=29 componentes (como en el caso del presente trabajo). En una situación como ésta, el número de parámetros a estimar para cada uno de los estados del modelo es de $2MD = 406$, correspondientes a MD medias y MD varianzas, adema de los M coeficientes de la mezcla. La estimación de tan elevado número de parámetros requiere de un conjunto de entrenamiento muy amplio. De otra forma, un conjunto de entrenamiento limitado deteriora el rendimiento de los modelos, al obtener estimaciones imprecisas de los parámetros [Rabiner85a, Huang89b], especialmente las matrices de covarianza de las gausianas de los estados.

Una posible alternativa para la reducción del número de parámetros a estimar es la de seleccionar un conjunto de gausianas compartido por todos los estados de los diferentes modelos. Una formulación en este sentido es la realizada para los modelos semicontínuos de Markov [Huang90a, Huang89b].

Desde el punto de vista de los modelos contínuos con mezcla de gausianas, los modelos semicontínuos comparten un conjunto común de gausianas en la composición de las mezclas correspondientes a las densidades de probabilidad de producción de observaciones de los estados de los modelos.

Desde el punto de vista de los modelos discretos, el diccionario VQ está modelado por un conjunto de gausianas, cada una de las cuales corresponde a un centro del diccionario, asociado a un símbolo del alfabeto discreto de símbolos observables, de forma que el proceso VQ genera un conjunto de probabilidades $f(x \mid O_j)$ del diccionario, correspondientes a cada uno de los símbolos o_j, para cada vector de observación x. De esta forma, la probabilidad de observación de un vector de parámetros x en un estado s_i de un modelo puede expresarse de forma análoga a la utilizada para la reestimación de los parámetros en los modelos contínuos (3.42),(3.43).

$$b_i(x) = \sum_{j=1}^{L} f(x \mid O_j) P(O_j \mid s_i) = \sum_{j=1}^{L} f(x \mid O_j) b_i(O_j) \tag{6.1}$$

donde $b_i(O_j)$ es la probabilidad discreta de observación del símbolo O_j en el estado s_i del modelo, y L es el número de centros en el diccionario VQ.

La utilización de (6.1) para el modelado de las probabilidades de observación de vectores contínuos de parámetros permite combinar las características de distorsión del diccionario VQ, modeladas por las funciones densidad de probabilidad $f(x \mid O_j)$, con las probabilidades discretas de producción de símbolos $b_i(O_j)$, en un contexto probabilístico.

De otro lado, (6.1) puede contemplarse como una mezcla de L gausianas, en la que las probabilidades discretas $b_i(O_j)$ juegan el papel de coeficientes de la mezcla. En la práctica, el número de componentes considerado en la expresión de la densidad de probabilidad de observación se limita a los M valores más significativos de $f(x \mid O_j)$, lo que reduce significativamente los requerimientos de cálculo del sistema.

6.2.1. FORMULAS DE EVALUACION Y REESTIMACION

Supuesto que las probabilidades de producción de observaciones para los SCHMM van a ser modeladas mediante una combinación lineal

$$b_i(x) = \sum_{j=1}^{L} f(x \mid O_j)\, b_i(O_j) \tag{6.2}$$

de los M centros del diccionario VQ con valores más significativos de $f(x \mid O_j)$, es necesario modificar las fórmulas de evaluación y reestimación de los modelos HMM discretos para contemplar esta nueva situación.

En cuanto a las fórmulas de evaluación, basta con sustituir las probabilidades discretas $b_i(O_t)$ por las densidades de probabilidad $b_i(x_t)$ calculadas según (6.2).

Para las fórmulas de reestimación, es necesario modificar además la expresión correspondiente a la reestimación de las probabilidades de producción de símbolos $b_i(O_j)$. Las nuevas ecuaciones de reestimación para éstas probabilidades se pueden expresar [Huang89b] en la forma siguiente

$$\bar{b}_i(O_j) = \frac{\sum_{t=1}^{T} \gamma_t(i,j)}{\sum_{k=1}^{M} \sum_{t=1}^{T} \gamma_t(i,k)} \tag{6.3a}$$

$$\gamma_t(i,j) = \left[\frac{\alpha_t(i)\,\beta_t(j)}{\sum_{k=1}^{M} \alpha_t(k)\,\beta_t(k)} \right] \cdot \left[\frac{b_i(O_j)\,f(x_t \mid O_j)}{\sum_{m=1}^{M} b_i(O_m)\,f(x_t \mid O_m)} \right] \tag{6.3b}$$

El resto de las fórmulas de reestimación permanecen inalteradas salvo la modificación necesaria en el cálculo de $\alpha_t(i)$ y $\beta_t(i)$ en las que las probabilidades $b_i(o_t)$ deben sustituirse por las densidades de probabilidad $b_i(x_t)$ evaluadas de acuerdo con (6.2).

También es posible formular relaciones para la reestimación de las medias y las varianzas del diccionario VQ en la forma siguiente:

$$\bar{\mu}_i = \frac{\sum_{v} \sum_{t=1}^{T} \gamma_t^v(i)\, x_t^v}{\sum_{v} \sum_{t=1}^{T} \gamma_t^v(i)} \tag{6.4a}$$

$$\bar{\Sigma}_i = \frac{\sum_{v} \sum_{t=1}^{T} \gamma_t^v(i)\,(x_t^v - \mu_i)^T\,(x_t^v - \mu_i)}{\sum_{v} \sum_{t=1}^{T} \gamma_t^v(i)} \tag{6.4b}$$

$$\gamma_t^v(i) = \sum_{j=1}^{M} \gamma_t^v(i,j) \tag{6.4c}$$

donde la suma en v denota suma sobre todas las secuencias de entrenamiento (de todas las palabras).

Tales fórmulas se han utilizado para la reestimación de medias y varianzas [Huang90], obteniendo moderados incrementos en el rendimiento del sistema para la reestimación de las medias; la reestimación de las varianzas, sin embargo, empeora el rendimiento del sistema. Por estos motivos, y por la complejidad computacional introducida por (6.4a-c), en la presente implementación no utilizaremos las formulas de reestimación

para las medias y varianzas de los centros del diccionario.

6.3. EL SISTEMA DE RECONOCIMIENTO CON MODELOS SCHMM

En la implementación del sistema de reconocimiento con modelos semicontínuos, se utilizó el diccionario VQ construido para la implementación del sistema con modelos discretos, y se estimaron las varianzas de los centros de dicho diccionario sobre la totalidad de los vectores del conjunto de entrenamiento.

En cuanto al entrenamiento de los modelos, se utilizaron los modelos discretos previamente construidos como modelos iniciales, y se reestimaron utilizando un algoritmo Baum-Welch similar al utilizado en el modelado HMM discreto, con las modificaciones citadas en la sección anterior para la reestimación de las probabilidades discretas de observación de símbolos y para las fórmulas de evaluación.

Elección del número de candidatos VQ

Para la determinación del número óptimo M de candidatos VQ considerados en la expresión de las densidades de probabilidad de producción de observaciones de los modelos, se realizaron experimentos para determinar éste sobre configuraciones multilocutor e independiente del locutor del sistema. Para el primero, se utilizaron dos repeticiones de cada palabra pronunciadas por todos los locutores como conjunto de entrenamiento, y la tercera para test. Para la configuración independiente del locutor, se utilizaron todas las repeticiones de los 26 primeros locutores como conjunto de entrenamiento, y las correspondientes a los 14 restantes como conjunto de test, manteniendo en las dos particiones el mismo número de locutores masculinos que femeninos. Los resultados se muestran en la figura 6.1, para una configuración con 128 símbolos y 10 estados por modelo. Similares resultados se obtienen para otros valores del número de símbolos considerado.

A la vista de estos resultados, se seleccionó el valor $M = 10$, para el número de candidatos VQ considerados. Valores superiores de M sólo reducen moderadamente el error o incluso lo incrementan, incrementando el coste computacional de los modelos.

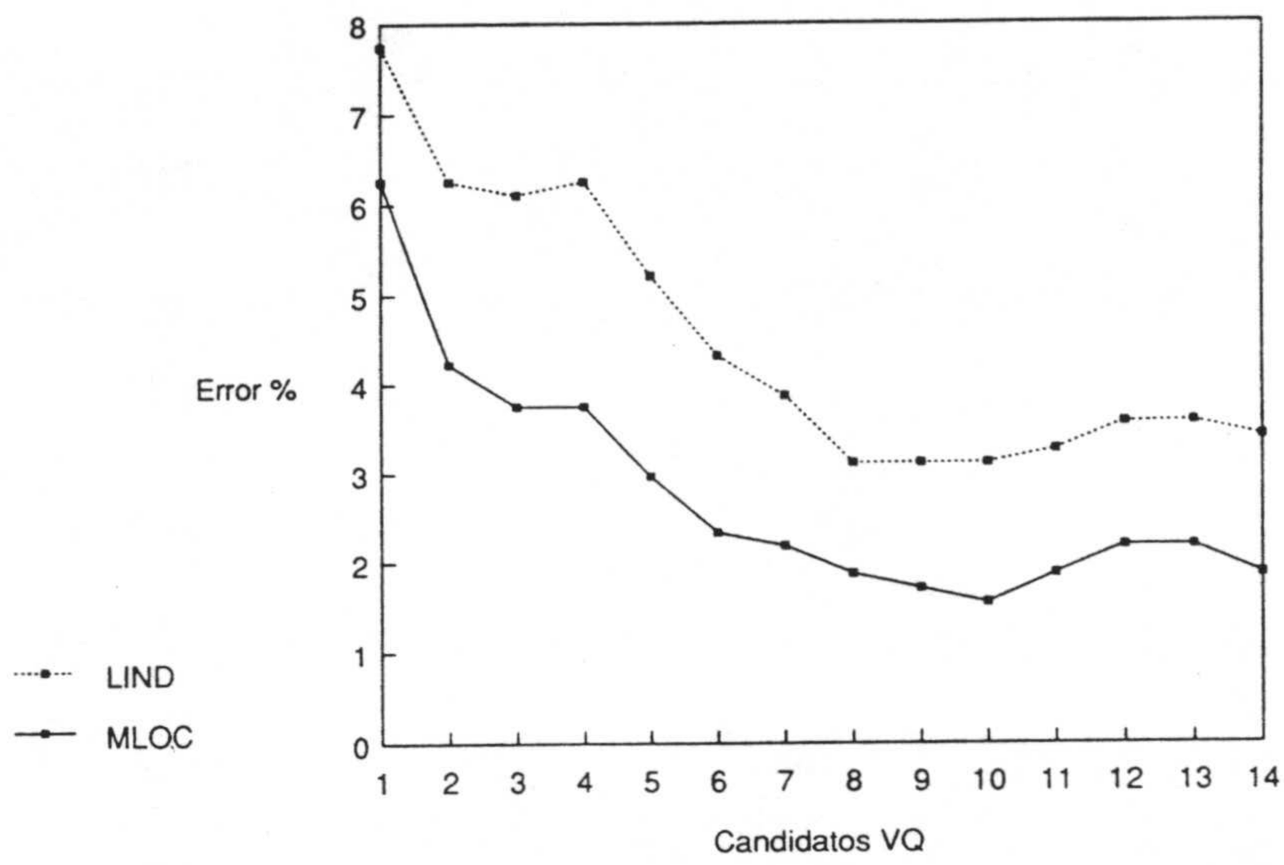

Figura 6.1. Error de reconocimiento frente a M para los modelos SCHMM

Resultados de reconocimiento

Una vez fijado el valor de M, se realizaron experimentos extensivos L1OUT y L4OUT sobre la base de datos, con los resultados mostrados en la tabla VIII. Estos resultados corresponden a una configuración del sistema de reconocimiento análoga a la utilizada en el caso de modelos discretos, incluyendo la adición de los logaritmos de las probabilidades de duración de estados con el mismo valor $W_{DUR}=0.5$ empleado para el modelado discreto.

Ns	L1OUT		L4OUT	
	DHMM	SCHMM	DHMM	SCHMM
64	3.08%	1.77%	4.37%	3.02%
128	1.93%	1.14%	3.59%	2.13%
256	1.36%	0.73%	2,81%	1.56%
512	0.99%	0.52%	2.03%	1.43%

Tabla VIII. Error de reconocimiento para los modelos DHMM y SCHMM.

Estos resultados muestran la efectividad del modelado HMM semicontínuo, al tener en cuenta múltiples candidatos en el proceso VQ. Es de notar, sin embargo, que este incremento en el rendimiento del sistema es a costa de un incremento en el coste computacional del mismo. La implementación descrita con modelos semicontínuos, para un mismo número de centros en el diccionario VQ que en el caso de modelos discretos, tiene un coste computacional del orden del doble debido a la necesidad de calcular los valores de las densidades de probabilidad de cada uno de los centros del diccionario. El cálculo de la distorsión de cuantización de un vector con un centro del diccionario requiere de la evaluación de D diferencias y D productos (D es el número de componentes del vector), mientras que el cálculo del valor de la densidad de probabilidad asociada a un centro del diccionario requiere D diferencias, D productos y D cocientes.

6.4. MODELOS HMM CON CUANTIZACION DEPENDIENTE

Tal y como se indicó en la introducción, varios trabajos sobre reconocimiento de palabras aisladas utilizando técnicas VQ con múltiples diccionarios mostraron que la distorsión de cuantización de una secuencia de vectores correspondiente a una palabra incógnita sobre un conjunto de diccionarios VQ de referencia construidos para cada una de las palabras de un determinado vocabulario puede utilizarse de forma efectiva para la clasificación de ésta. En esta sección resumiremos las aproximaciones utilizadas para la implementación de tales sistemas, y describiremos los modelos de Markov con cuantización dependiente MVQHMM, en los que la distorsión de cuantización de las secuencias incógnitas con diccionarios VQ específicos para cada modelo se combina con la probabilidad de generación de las secuencias de símbolos obtenidas de dichas cuantizaciones para obtener una probabilidad de generación compuesta que se puede utilizar para la clasificación de secuencias de vectores correspondientes a palabras incógnita.

6.4.1. RECONOCIMIENTO DE VOZ CON MULTIPLES DICCIONARIOS VQ

Los primeros trabajos sobre reconocimiento de voz mediante la utilización de las distorsiones de cuantización sobre multiples diccionarios VQ correspondientes a diferentes palabras de un vocabulario se deben a Buzo [Buzo82] y Shore [Shore82], que describieron

sistemas de reconocimiento basados en una clasificación geométrica de secuencias de vectores de características espectrales de señales de voz sin ningún tipo de alineamiento temporal. En tales sistemas, cada una de las palabras del vocabulario se caracteriza por un diccionario VQ, cuyos centros representan diferentes tipos de envolventes espectrales de los segmentos de señal correspondientes a dicha palabra. La clasificación de secuencias incógnita se realiza en base a la cuantización de los vectores de características con cada uno de los diccionarios de las diferentes palabras del vocabulario, seleccionando aquel para el que la distorsión media de cuantización es menor.

Esta aproximación es básicamente la misma que se utiliza en programación dinámica (DTW), salvo por el hecho de que no se tiene en cuenta ningún tipo de alineamiento temporal entre los segmentos de la palabra incógnita y los prototipos. Esto hace que los requerimientos de cálculo sean mucho menores para estos sistemas que para los basados en DTW.

Incluso sin la utilización de ningún tipo de alineamiento temporal, estos sistema ofrecen tasas de reconocimiento buenas. Así, Shore obtuvo un 98.8% de reconocimiento sobre un vocabulario de 20 palabras, independiente del locutor para locutores masculinos, y Bergh [Bergh85] obtuvo un 91.4% de reconocimiento para una base de datos de dígitos aislados independiente del locutor con locutores masculinos y femeninos. Sin embargo, cuando el sistema se experimentó con un vocabulario más grande, el rendimiento decayó bruscamente. Para una base de datos con 129 palabras independiente del locutor, el sistema únicamente produjo un 65.4% de reconocimiento, lo que indica que cuando el tamaño del vocabulario es grande, y existen palabras con características espectrales similares, la mera clasificación de los patrones espectrales no es suficiente para el reconocimiento, y es necesario incorporar información sobre el secuenciamiento temporal de los prototipos espectrales, para aumentar la discriminación del sistema de reconocimiento. Para tener en cuenta información sobre el alineamiento temporal de los segmentos de señal considerados en el sistema, se propusieron varias alternativas que pasamos a describir a continuación.

Utilización de un postprocesador DTW

Pan [Pan85] y Furui [Furui88] incorporaron un postprocesador basado en DTW, de forma que las distorsiones VQ se utilizaron en un preprocesador para la eliminación de candidatos improbables, mientras que los candidatos aceptados por éste eran procesados

por un algoritmo DTW standard que emitía las decisiones finales.

Diccionarios multisección

Burton [Burton85b] modificó su sistema de reconocimiento dividiendo cada palabra en un número fijo de secciones temporales de igual duración, y diseñando un diccionario VQ para cada una de las secciones temporales obtenidas para cada palabra del vocabulario. Al conjunto de diccionarios correspondientes a las diferentes secciones de una misma palabra lo denominó diccionario multisección.

De esta forma, la cuantización vectorial de los segmentos de las palabras incógnita se realiza dividiendo éstas uniformemente en tantas secciones como las del diccionario multisección de la palabra del vocabulario considerada, y cuantizando los segmentos en cada una de las secciones de la palabra incógnita, con el diccionario de la sección correspondiente. De esta forma, el sistema realiza un cierto tipo de alineamiento temporal entre las secciones correspondientes a las palabras incógnita y las secciones del diccionario VQ. Para esta nueva configuración del sistema, y sobre la misma base de datos independiente del locutor de dígitos aislados utilizada por Bergh, se obtuvo un 98.7% de reconocimiento.

Distribución temporal de observación de símbolos

Bergh [Bergh85] diseñó un postprocesador para el sistema de reconocimiento basado en la evaluación de las probabilidades temporales de observación de los símbolos obtenidos en el proceso de cuantización vectorial. De esta forma, el sistema, además de los diccionarios VQ correspondientes a cada una de las palabras del vocabulario, utiliza una estimación de las distribuciones temporales de probabilidad de observación de cada uno de los símbolos (centros del diccionario) obtenidos en el proceso de cuantización vectorial. La estimación de estas probabilidades temporales de observación se realiza a través de una estadística de los instantes de tiempo (normalizados a la duración de las palabras) en los que son observados los diferentes centros de los diccionarios en las cuantizaciones de las palabras.

En la fase de reconocimiento, las palabras incógnita son cuantizadas con los diferentes diccionarios, y posteriormente se evalua la probabilidad temporal de observación

de la secuencia de símbolos obtenida, de forma que las distorsiones medias de cuantización se combinan linealmente, con un peso relativo ajustado experimentalmente, con los logaritmos de las probabilidades de observación temporales para obtener una distorsión total que es utilizada en la clasificación de las palabras incógnita. Esta combinación se realiza de forma que las palabras cuyas vectorizaciones presentan distribuciones temporales de símbolos que no concuerdan con la estimación correspondiente a un modelo determinado son penalizadas aumentando el valor de la distorsión total.

Para esta nueva configuración del sistema de reconocimiento, Bergh mostró incrementos significativos en el tanto por ciento de reconocimiento obteniendo un 97.2% de reconocimiento para la base de datos de dígitos aislados, y un 88.1% para la base de datos de 129 palabras antes citada. Este último resultado muestra cómo la adición de información sobre alineamiento temporal es especialmente efectiva cuando el tamaño del vocabulario el grande; para el vocabulario de 129 palabras, la tasa de reconocimiento aumenta de un 65.4% a un 88.1%.

6.4.2. MODELOS MVQHMM

Los resultados expuestos en la sección anterior muestran que las distorsiones de cuantización de los segmentos correspondientes a una palabra incógnita con un conjunto de diccionarios VQ correspondientes a las diferentes palabras del vocabulario es una información útil para el reconocimiento de palabras aisladas. Sin embargo, es necesaria la incorporación de información sobre el secuenciamiento temporal de los símbolos obtenidos en la cuantización vectorial para aumentar la discriminación del sistema de reconocimiento. En la presente sección presentaremos una nueva variante del modelado discreto de Markov, que permite la integración natural de las informaciones relativas a las distorsiones de cuantización y al secuenciamiento temporal de los símbolos obtenidos en dicha cuantización. En adelante, a este tipo de modelado lo denominaremos, como ya indicamos anteriormente, modelado MVQHMM.

Un modelo MVQHMM de una palabra está compuesto por un diccionario VQ cuyos centros representan los posibles valores de los vectores de parámetros extraidos para dicha palabra, y que a su vez constituyen el conjunto discreto de observaciones para un proceso de Markov modelado por un HMM discreto. De esta forma, el diccionario VQ modela el conjunto de posibles valores para los vectores de características de la palabra, mientras que

el HMM discreto modela el secuenciamiento temporal de los prototipos espectrales representados por los centros del diccionario VQ.

Cuando un MVQHMM es utilizado para la clasificación de una secuencia de vectores X_1^T correspondiente a una palabra incógnita, se utiliza el diccionario VQ para cuantizar dicha secuencia y obtener una secuencia de símbolos O_1^T, y el valor de la distorsión de cuantización. Entonces, el modelo HMM se utiliza para evaluar la probabilidad de generación de la secuencia de símbolos O_1^T, que se combina con la distorsión de cuantización para obtener un valor que se utiliza para la clasificación de dicha secuencia de vectores.

6.5. DERIVACION DE LOS MVQHMM

En la sección anterior hemos introducido el concepto de modelo de Markov con cuantización dependiente, en esta sección realizaremos una derivación formal de dichos modelos, y describiremos la forma en que las distorsiones de cuantización y las probabilidades de generación se combinan para obtener la regla de decisión utilizada en el proceso de clasificación.

Para esta derivación partiremos de una formulación HMM para el cálculo de la probabilidad de generación de una secuencia de vectores contínuos observables X_1^T. Para esta secuencia, podemos escribir la siguiente relación para la probabilidad de generación a posteriori dado un modelo oculto de Markov λ

$$P(X_1^T \mid \lambda) = \sum_{S^T} P(X_1^T \mid S_1^T, \lambda) P(S_1^T \mid \lambda) \tag{6.5}$$

donde S^T denota la sumatoria sobre todas las posibles secuencias de estados para el modelo. Los dos factores de la sumatoria se pueden expresar en la forma siguiente

$$P(X_1^T \mid S_1^T, \lambda) = \prod_{t=1}^{T} P(x_t \mid s_t, \lambda) \tag{6.6a}$$

$$P(S_1^T \mid \lambda) = P(s_1 \mid \lambda) \prod_{t=1}^{T} P(s_t \mid s_{t-1}, \lambda) \tag{6.6b}$$

Asumiendo que las funciones densidad de probabilidad de producción de observaciones $P(x_t \mid s_t, \lambda)$ pueden expresarse como combinación lineal de funciones densidad de probabilidad, podemos reescribir (6.6a) en la forma

$$P(x_t \mid s_t, \lambda) = \sum_{o_t \in V(s_t, \lambda)} P(x_t \mid o_t, s_t, \lambda) P(o_t \mid s_t, \lambda) \tag{6.7}$$

donde la suma en o_t se extiende a un conjunto de clases $V(s_t, \lambda)$ caracterizadas por funciones densidad de probabilidad específicas para cada estado s_t de cada modelo λ. Tal descomposición es la utilizada en el modelado HMM contínuo en el que las densidades de probabilidad $P(s_t \mid o_t, s_t, \lambda)$ son modeladas utilizando gausianas multivariadas. En esta situación, las probabilidades $P(o_t \mid s_t, \lambda)$ juegan el papel de coeficientes de la mezcla de gausianas.

A partir de esta formulación es sencillo derivar la correspondiente a los modelos semicontínuos sin más que reescribir (6.7) eliminando la dependencia de $P(s_t \mid o_t, s_t, \lambda)$ con s_t y λ

$$P(x_t \mid s_t, \lambda) = \sum_{o_t \in V} P(x_t \mid o_t) P(o_t \mid s_t, \lambda) \tag{6.8}$$

de forma que ahora, la sumatoria en o_t se extiende a un conjunto V de clases caracterizado por funciones densidad de probabilidad comunes a todos los estados de los diferentes modelos.

La formulación correspondiente a los modelos discretos se deriva directamente de la anterior sin más que reducir la sumatoria en o_t a un único término en la forma

$$P(x_t \mid s_t, \lambda) = P(x_t \mid o_t^*) P(o_t^* \mid s_t, \lambda) \tag{6.9a}$$

$$o_t^* = \underset{o_t \in V}{\operatorname{argmax}} \{ P(x_t \mid o_t) \} \tag{6.9b}$$

En esta aproximación se ha supuesto que las clases a las que pertenecen los símbolos o_t son disjuntas (o levemente solapadas), de forma que la sustitución propuesta no representa pérdida de información significativa.

Utilizando las expresiones (6.5), (6.6) y (6.9), podemos escribir la probabilidad de generación a posteriori de la secuencia de vectores X_1^T en la forma siguiente

$$P(X_1^T \mid S_1^T, \lambda) = \prod_{t=1}^{T} P(x_t \mid s_t, \lambda) \tag{6.10a}$$

$$= \prod_{t=1}^{T} P(x_t \mid o_t^*) P(o_t^* \mid s_t, \lambda) \tag{6.10b}$$

$$= \left[\prod_{t=1}^{T} P(x_t \mid o_t^*) \right] \left[\prod_{t=1}^{T} P(o_t^* \mid s_t, \lambda) \right] \tag{6.10c}$$

$$= P(X_1^T \mid O^{*}{}_1^T) P(O^{*}{}_1^T \mid S_1^T, \lambda) \tag{6.10d}$$

$$P(X_1^T \mid \lambda) = \sum_{S^T} P(X_1^T \mid S_1^T, \lambda) P(S_1^T \mid \lambda) \tag{6.11a}$$

$$= \sum_{S^T} P(X_1^T \mid O^{*}{}_1^T) P(O^{*}{}_1^T \mid S_1^T, \lambda) P(S_1^T \mid \lambda) \tag{6.11b}$$

$$= P(X_1^T \mid O^{*}{}_1^T) \sum_{S^T} P(O^{*}{}_1^T \mid S_1^T, \lambda) P(S_1^T \mid \lambda) \tag{6.11c}$$

$$= P(X_1^T \mid O^{*}{}_1^T) P(O^{*}{}_1^T \mid \lambda) \tag{6.11d}$$

Tal y como se muestra en (6.11d), la probabilidad de generación de la secuencia de vectores X_1^T puede expresarse como producto de dos probabilidades, la primera $P(X_1^T \mid O^{*}{}_1^T)$ es la probabilidad de la cuantización de X_1^T en la secuencia de símbolos $O^{*}{}_1^T$ de máxima probabilidad, y la segunda $P(O^{*}{}_1^T \mid \lambda)$ es la probabilidad de generación de la

secuencia de símbolos dado el modelo discreto de Markov λ.

Dado que $P(X_1^T \mid O^*{}_1^T)$ es constante para una secuencia de vectores dada, en la implementación del modelado HMM discreto para reconocimiento, sólo es necesario evaluar la probabilidad de generación de la cadena de símbolos $P(O^*{}_1^T \mid \lambda)$. La probabilidad $P(X_1^T \mid O^*{}_1^T)$ sólo es tenida en cuenta en el diseño del cuantizador del sistema, de forma que ésta es maximizada sobre el conjunto de secuencias de entrenamiento del mismo en la fase de construcción del diccionario VQ.

Una vez que se ha mostrado como derivar los diferentes tipos de modelos HMM propuestos hasta ahora, pasamos a derivar una expresión para la probabilidad de generación para los modelos MVQHMM. Para ésto, repetiremos el mismo proceso seguido para llegar a la probabilidad de generación para los modelos discretos, pero relajaremos la condición de que todos los modelos compartan el mismo conjunto de símbolos; así, la relación (6.7) puede escribirse en la forma

$$P(x_t \mid s_t, \lambda) = \sum_{o_t \in V(\lambda)} P(x_t \mid o_t, \lambda)\, P(o_t \mid s_t, \lambda) \tag{6.12}$$

donde ahora la suma en o_t se extiende a un conjunto de clases $V(\lambda)$ específico para el modelo λ considerado.

Introduciendo la condición de clases disjuntas podemos reescribir esta ecuación en la forma siguiente

$$P(x_t \mid s_t, \lambda) = P(x_t \mid o_t^*, \lambda)\, P(o_t^* \mid s_t, \lambda) \tag{6.13a}$$

$$o_t^* = \underset{o_t \in V(\lambda)}{\operatorname{argmax}} \{ P(x_t \mid o_t, \lambda) \} \tag{6.13b}$$

Utilizando (6.13), (6.5) y (6.6) podemos escribir las relaciones

$$P(X_1^T \mid S_1^T, \lambda) = \prod_{t=1}^{T} P(x_t \mid s_t, \lambda) \tag{6.14a}$$

$$= \prod_{t=1}^{T} P(x_t \mid o_t^*, \lambda) P(o_t^* \mid s_t, \lambda) \tag{6.14b}$$

$$= \left[\prod_{t=1}^{T} P(x_t \mid o_t^*, \lambda)\right] \left[\prod_{t=1}^{T} P(o_t^* \mid s_t, \lambda)\right] \tag{6.14c}$$

$$= P(X_1^T \mid O^{*}{}_1^T, \lambda) P(O^{*}{}_1^T \mid S_1^T, \lambda) \tag{6.14d}$$

$$P(X_1^T \mid \lambda) = \sum_{S^T} P(X_1^T \mid S_1^T, \lambda) P(S_1^T \mid \lambda) \tag{6.15a}$$

$$= \sum_{S^T} P(X_1^T \mid O^{*}{}_1^T, \lambda) P(O^{*}{}_1^T \mid S_1^T, \lambda) P(S_1^T \mid \lambda) \tag{6.15b}$$

$$= P(X_1^T \mid O^{*}{}_1^T, \lambda) \sum_{S^T} P(O^{*}{}_1^T \mid S_1^T, \lambda) P(S_1^T \mid \lambda) \tag{6.15c}$$

$$= P(X_1^T \mid O^{*}{}_1^T, \lambda) P(O^{*}{}_1^T \mid \lambda) \tag{6.15d}$$

La relación (6.13b) establece la condición de cuantización vectorial de la secuencia de vectores de entrada con el diccionario específico del modelo λ considerado, y la relación (6.15d) indica cómo se componen las probabilidades de cuantización $P(X_1^T \mid O^{*}{}_1^T, \lambda)$ con las probabilidades de generación $P(O^{*}{}_1^T \mid \lambda)$.

En este caso, la probabilidad de cuantización de la secuencia de vectores ya no es constante, sino que depende del modelo considerado, y por lo tanto es necesario tenerla en cuenta en el cálculo de la probabilidad total de generación $P(X_1^T \mid \lambda)$. Esta probabilidad es la que tiene en cuenta las características de distorsión de los diferentes diccionarios VQ correspondientes a los modelos considerados.

6.5.1. ENTRENAMIENTO DE LOS MODELOS MVQHMM

En un contexto de estimación de máxima probabilidad, el entrenamiento de un modelo MVQHMM necesita de la maximización de $P(X_1^T \mid \lambda)$, lo que implica la

maximización conjunta de $P(X_1^T \mid O^*{}_1^T, \lambda)$ y $P(O^*{}_1^T \mid \lambda)$. Dado que ésto requeriría un gran volumen de cálculo, utilizaremos la misma aproximación que en caso del modelado HMM discreto, es decir, la maximización se realiza en dos fases, en la primera de ellas se obtiene un conjunto de clases (diccionario) que maximiza la probabilidad de cuantización, y una vez obtenido éste, se maximiza la probabilidad de generación de las secuencias de símbolos (clases) obtenidas de la cuantización de las secuencias de vectores contínuos con el diccionario previamente construido.

Para la estimación de las probabilidades de cuantización, utilizaremos un modelado gausiano de los centros del diccionario de tal forma que asumiremos que las matrices de covarianza de los centros del diccionario son de la forma

$$\Sigma_{j\lambda} = \sigma_\lambda^2 I \tag{6.16}$$

donde I es la matriz identidad, y σ_λ^2 es la varianza de los componentes de los vectores de características. Esta formulación permite simplificar el proceso de cuantización vectorial, y de evaluación de las probabilidades de cuantización. Además, al asumir igual varianza para todos los componentes de los vectores, se reduce el número de parámetros a estimar para las matrices de covarianza.

Con estas suposiciones, las funciones densidad de probabilidad de los centros del diccionario del modelo λ toman la forma

$$P(x \mid o_j, \lambda) = (2\pi)^{-p/2} (\sigma_\lambda^2)^{-p/2} \exp\left\{ \frac{-1}{2\sigma_\lambda^2} \| x - \mu_{j\lambda} \|^2 \right\} \tag{6.17}$$

donde p es el número de componentes del vector x y $\mu_{j\lambda}$ es el vector media del centro o_j del modelo λ.

Bajo estas condiciones, el logaritmo de la probabilidad a posteriori de generación de un vector x dado un centro o_j de un modelo λ, es directamente proporcional a la distancia euclídea entre el vector x y el vector media del centro considerado $\mu_{j\lambda}$, con lo que el proceso de asignación de símbolos a vectores según un criterio de máxima probabilidad a posteriori, se reduce a la selección del centro más próximo del diccionario considerado.

Para la completa caracterización de las funciones densidad de probabilidad sólo es necesaria la determinación de los vectores media de los centros del diccionario, que se puede realizar mediante un algoritmo jerárquico de agrupamiento como el utilizado para la construcción de los diccionarios para los modelos DHMM, y a la determinación de las varianzas σ_λ^2 de cada uno de los diccionarios de los modelos.

Una vez construido el diccionario VQ del modelo, las secuencias de vectores de entrenamiento son cuantizadas obteniendo las secuencias de símbolos correspondientes. Estas secuencias son utilizadas en el entrenamiento del HMM correspondiente al modelo MQHMM de la palabra.

6.5.2. CLASIFICACION DE SECUENCIAS CON LOS MVQHMM

En el proceso de clasificación de una secuencia de vectores incógnita X_1^T, se procede de forma análoga al caso del modelado HMM discreto, con la salvedad de que el proceso de cuantización vectorial se realiza con los diccionarios previamente construidos para cada uno de los modelos. A partir de la secuencia de símbolos O_1^T obtenida[1] en el proceso de cuantización vectorial, se evalua la probabilidad a posteriori de generación con el modelo HMM discreto asociado al MVQHMM. Por último, la suma de los logaritmos de las probabilidades de cuantización y generación componen la probabilidad total de generación de la secuencia de vectores.

En la figura 6.2 se muestra un esquema del sistema de reconocimiento basado en modelos con cuantización dependiente. En ésta, se ha añadido un bloque correspondiente a la evaluación de las probabilidades de duración de estados tal y como se utilizó para los modelos DHMM y SCHMM, junto con el conjunto de probabilidades de duración de los estados de cada modelo.

---

[1]En adelante, omitiremos el superíndice * entendiendo que los símbolos producidos durante la cuantización vectorial corresponden a los de máxima probabilidad a posteriori para el diccionario considerado.

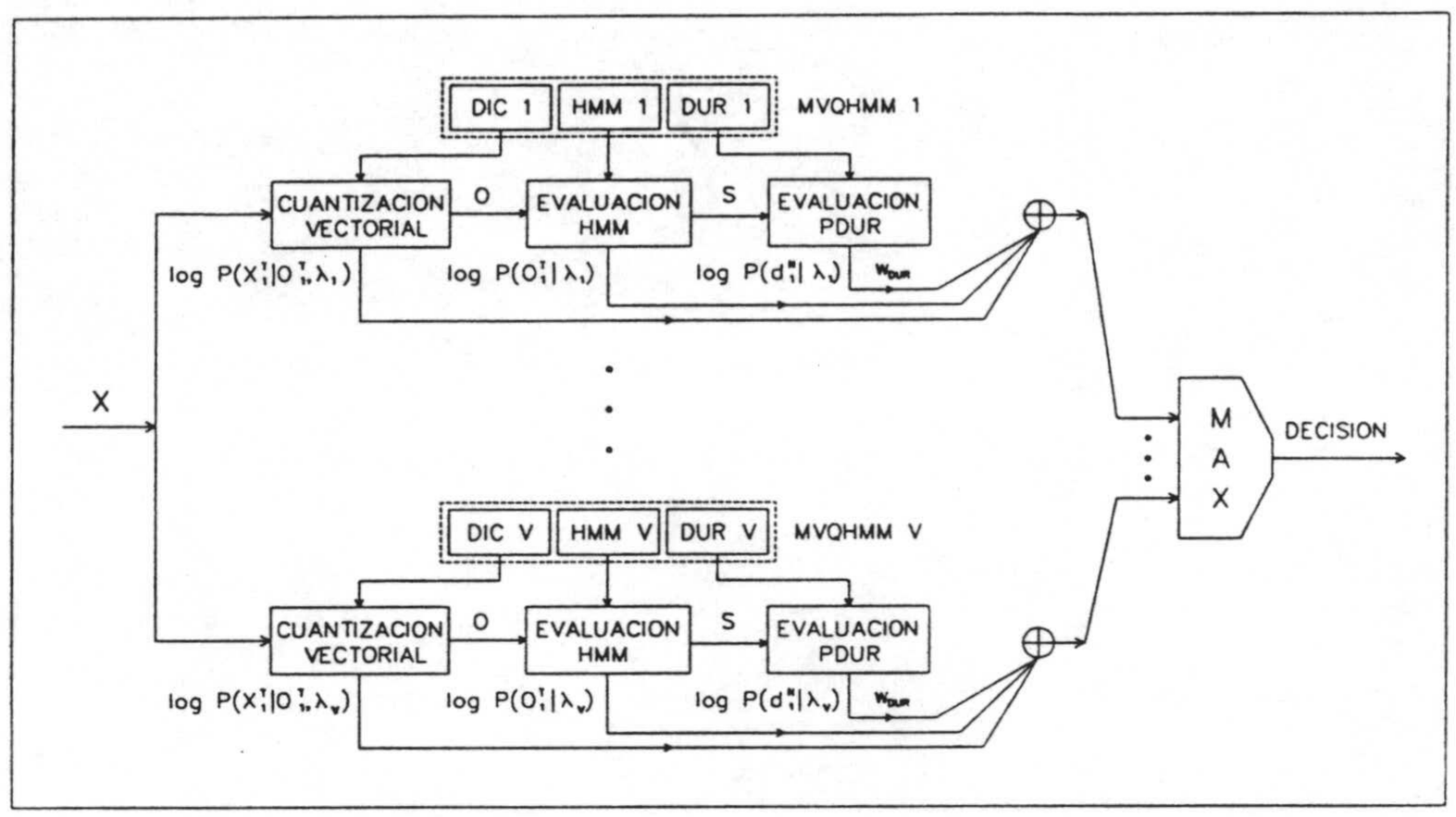

Figura 6.2. Esquema del sistema de reconocimiento con modelos MVQHMM

Evaluación de las probabilidades de cuantización

De acuerdo con (6.15d) y (6.17) podemos escribir, para las probabilidades logarítmicas de generación, las relaciones

$$\log P(X_1^T \mid \lambda) = \log P(X_1^T \mid O_1^T, \lambda) + \log P(O_1^T \mid \lambda) \tag{6.18}$$

$$\log P(X_1^T \mid O_1^T, \lambda) = -\,T\,\frac{p}{2}\log 2\pi \;-\,T\,\frac{p}{2}\log \sigma_\lambda^2 \;-\,\frac{1}{2\sigma_\lambda^2}\sum_{t=1}^{T}\| x_t - \mu_{o_t\lambda}\|^2 \tag{6.19}$$

donde $\mu_{o_t\lambda}$ es la media de la clase o_t asociada al vector x_t en el sentido de la cuantización vectorial.

Si definimos la distorsión media de cuantización de la secuencia de vectores X_1^T, con el diccionario del modelo λ, en la forma

$$D_\lambda(X_1^T) = \frac{1}{T}\sum_{t=1}^{T} \| x_t - \mu_{o_t,\lambda} \|^2 \tag{6.20}$$

entendiendo que el símbolo o_t está unívocamente asociado al vector x_t mediante el proceso de cuantización vectorial, y dividiendo (6.19) por la longitud T de la secuencia de vectores y sustituyendo (6.20) obtenemos la siguiente expresión para la probabilidad logarítmica media de cuantización

$$\frac{1}{T}\log P(X_1^T \mid O_1^T, \lambda) = -\frac{p}{2}\log 2\pi - \frac{p}{2}\log \sigma_\lambda^2 - \frac{D_\lambda(X_1^T)}{2\sigma_\lambda^2} \tag{6.21}$$

El valor σ_λ^2 puede estimarse teniendo en cuenta que el valor esperado de la distorsión de cuantización de los vectores de una palabra sobre el diccionario VQ correspondiente $\overline{D}_\lambda$ está relacionado con la varianza en la forma siguiente

$$\overline{D}_\lambda = p\ \sigma_\lambda^2 \tag{6.22}$$

utilizando (6.22) podemos reescribir (6.21) en la forma

$$\frac{1}{T}\log P(X_1^T \mid O_1^T, \lambda) = -\frac{p}{2}\log 2\pi - \frac{p}{2}\log\left[\frac{\overline{D}_\lambda}{p}\right] - \frac{p}{2}\frac{D_\lambda(X_1^T)}{\overline{D}_\lambda} \tag{6.23}$$

Nótese que si utilizamos (6.23) como regla de decisión para la clasificación de secuencias de vectores correspondientes a palabras de un determinado vocabulario, sólo es necesaria la evaluación de las distorsiones medias de cuantización $D_\lambda(X_1^T)$. Esta regla de decisión coincide con la utilizada en los trabajos descritos en la sección 6.4.1 si asumimos que las distorsiones medias de cuantización $\overline{D}_\lambda$ son iguales para todos los diccionarios. En (6.23), estas distorsiones medias modelan las características de distorsión de los diferentes diccionarios.

Por último, la probabilidad de generación se modifica con las probabilidades de duración de estados en la misma forma que se indicó para los modelos DHMM y SCHMM.

6.6. EL SISTEMA DE RECONOCIMIENTO CON MODELOS MVQHMM

En la implementación del sistema de reconocimiento mostrado esquemáticamente en la figura 6.2, se han utilizado los mismos algoritmos para construcción de diccionarios y entrenamiento de modelos HMM que en el caso de los modelos discretos previamente descritos en el capítulo 5.

Para la construcción de los diccionarios se utilizó un algoritmo jerárquico K-medias, y para el entrenamiento de los modelos de Markov se utilizó un algoritmo de reestimación Baum-Welch con iniciación por segmentación lineal, sobre las secuencias de símbolos generadas por el proceso de cuantización vectorial de las secuencias de vectores correspondientes a cada palabra, con el diccionario VQ del modelo correspondiente.

Los parámetros del sistema de reconocimiento, tanto los relativos a la construcción de los diccionarios como los correspondientes a los modelos discretos de Markov, se fijaron a los mismos valores utilizados en la implementación del sistema de reconocimiento basado en modelos discretos descrito en el capítulo 5.

6.6.1. COMPOSICION DE PROBABILIDADES

En primer lugar se realizaron experimentos L1OUT y L4OUT para determinar si el peso óptimo en la composición de las probabilidades de cuantización y generación es efectivamente el peso unidad, tal y como se dedujo en las secciones precedentes; para ésto se utilizó una composición de probabilidades de la forma

$$\frac{1}{T}\log P(X_1^T \mid \lambda) = \mu\left[\frac{1}{T}\log P(X_1^T \mid O_1^T, \lambda)\right] + (1-\mu)\left[\frac{1}{T}\log P(O_1^T \mid \lambda)\right]$$

fijando el valor W_{DUR} para la composición de la probabilidad de duración de estados a cero, con lo que ésta no se tuvo en cuenta. Se varió μ en el intervalo [0,1] obteniendo los resultados mostrados en la figura 6.3. Los resultados muestran que, para diferentes configuraciones del sistema correspondientes a diferente número de centros en los diccionarios VQ, el valor $\mu=0.5$ está siempre cercano al mínimo de error en las curvas de composición de probabilidades, lo que corresponde a pesos unitarios en la composición. Este resultado está de acuerdo con la regla de composición de probabilidades deducida en

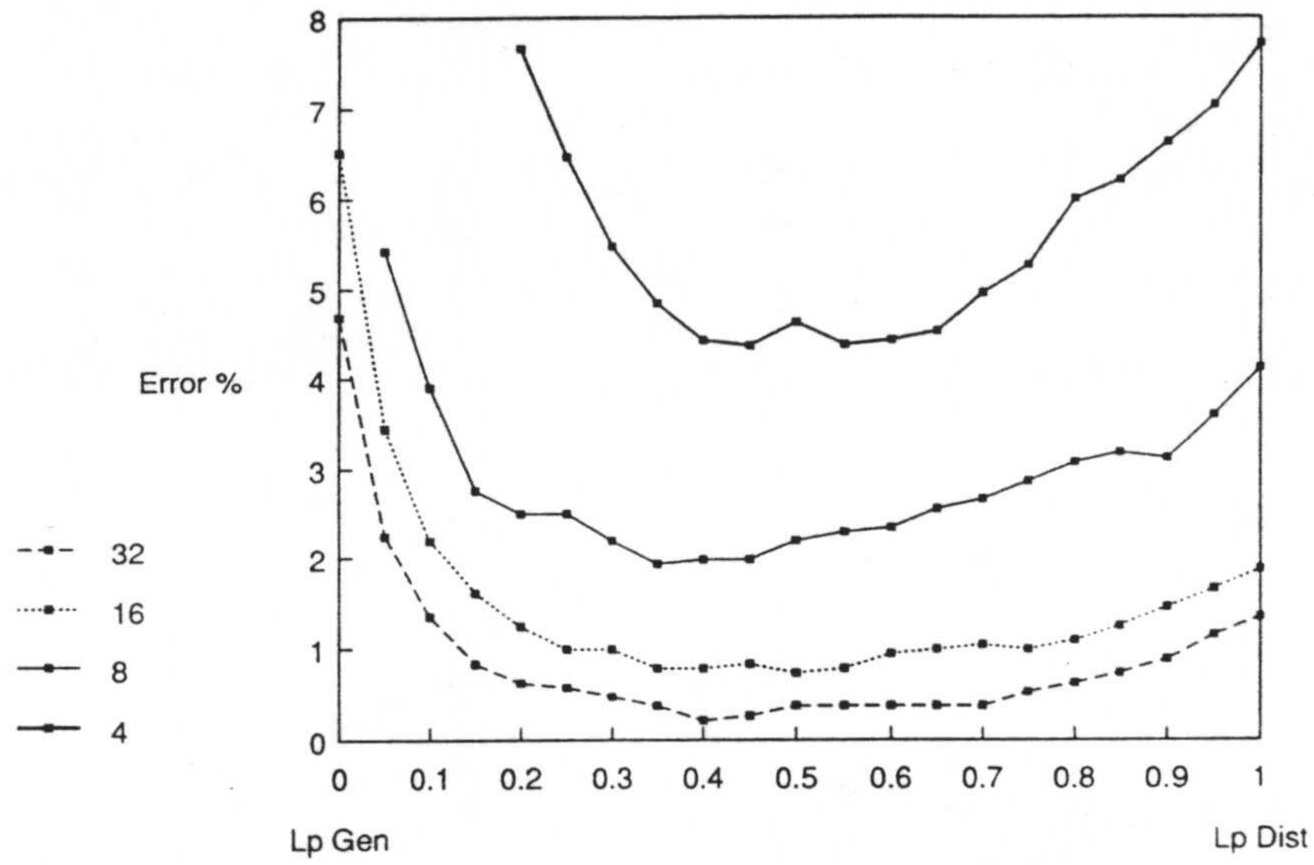

Figura 6.3a. Composición de probabilidades, experimento L1OUT

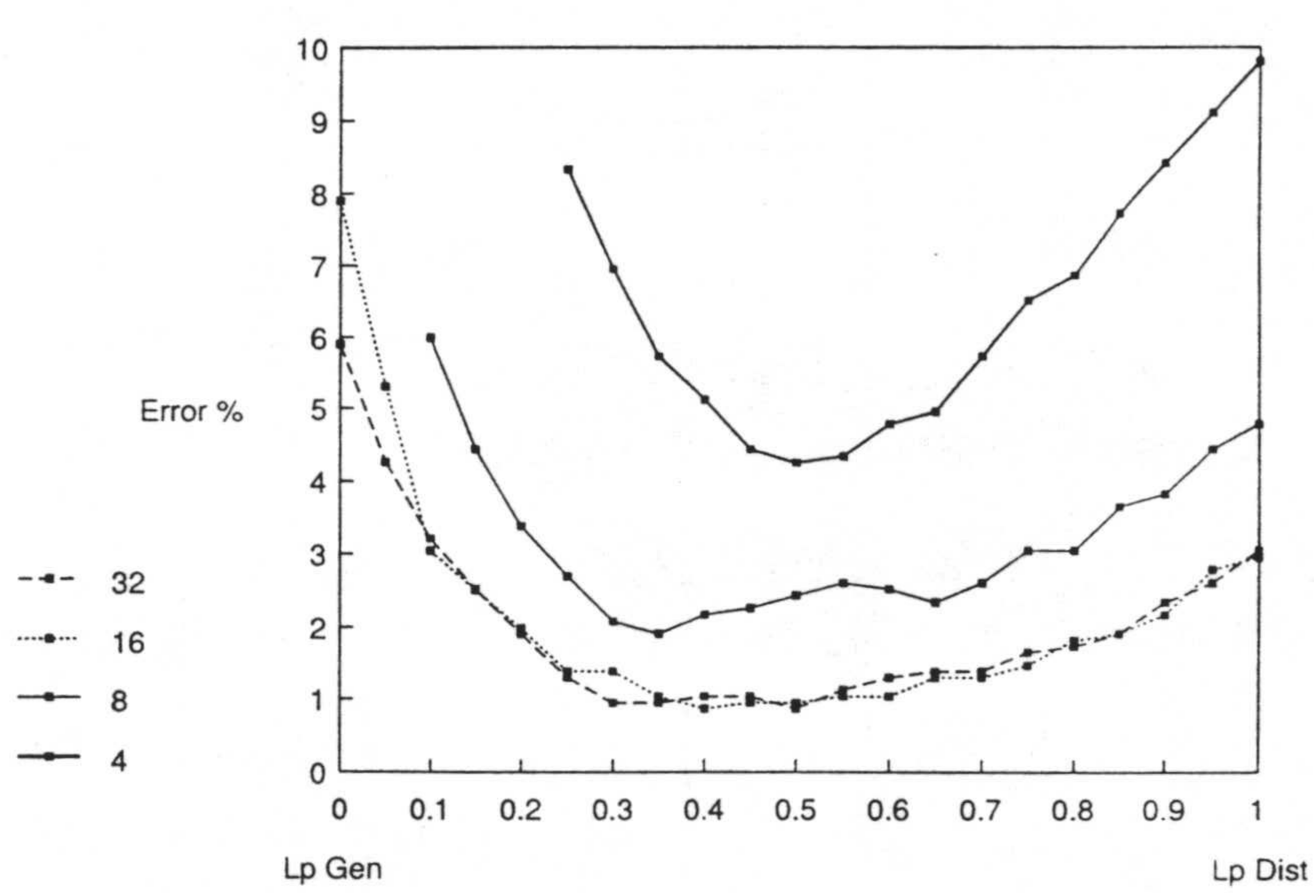

Figura 6.3b. Composición de probabilidades, experimento L4OUT

la sección anterior.

En cuanto al peso para la composición de las probabilidades de duración de estados, se realizaron experimentos obteniendo como resultado que el valor $W_{DUR} = 0.5$, utilizado

para los modelos DHMM y SCHMM, es también adecuado para esta nueva configuración del sistema.

6.6.2. RESULTADOS DE RECONOCIMIENTO

En la tabla IX se muestran los resultados de reconocimiento obtenidos para el sistema descrito, con modelos MVQHMM.

Ns	L1OUT			L4OUT		
	$P_{GEN}+P_{DUR}$	P_{DIST}	P_{TOT}	$P_{GEN}+P_{DUR}$	P_{DIST}	P_{TOT}
4	25.47%	7.71%	4.63%	28.56%	9.81%	4.25%
8	8.90%	4.12%	2.19%	13.63%	4.78%	2.43%
16	6.51%	1.87%	0.73%	7.90%	2.95%	0.95%
32	4.69%	1.35%	0.37%	5.90%	3.04%	0.87%

Tabla IX. Error de reconocimiento para los modelos MVQHMM.

En esta tabla, la columna etiquetada Ns se refiere al número de centros para cada uno de los diccionarios VQ de los modelos. La columna etiquetada $P_{GEN}+P_{DUR}$ corresponde a la utilización de la combinación de las probabilidades de generación de los modelos discretos y las probabilidades de duración de estados, la columna P_{DIST} se refiere a la utilización de las probabilidades de cuantización de las secuencias de vectores, y la columna P_{TOT} se refiere a los resultados cuando las tres probabilidades son utilizadas conjuntamente.

Composición de probabilidades

Estos resultados muestran la efectividad de la probabilidad de cuantización como parámetro de reconocimiento, de hecho, cuando únicamente se utiliza ésta como criterio de clasificación, ofrece menor tasa de error que cuando se utiliza únicamente la combinación de las probabilidades de generación de símbolos y de duración de estados. Como ya indicamos anteriormente, esta situación corresponde a un criterio de clasificación en el que no se tiene en cuenta ningún tipo de información sobre el alineamiento temporal de las secuencias de vectores, pero incluso así, la tasa de error obtenida con un número suficiente

de centros en los diccionarios VQ es similar a la obtenida con un modelado discreto. Las diferencias son menores del 1% para 16 centros en cada uno de los diccionarios VQ de las palabras del vocabulario, y 256 centros en el diccionario de los modelos discretos.

En cuanto al rendimiento obtenido cuando se utiliza únicamente la información relativa a las probabilidades de generación de símbolos y duración de estados (información suministrada por los modelos discretos asociados a los MVQHMM), columna $P_{GEN} + P_{DUR}$ de la tabla IX, se comprueba que el error es mucho más elevado que en el caso de los modelos discretos y semicontínuos. Esto es debido a que se ha eliminado, en el proceso de cuantización con diccionarios independientes para cada palabra, la información relativa a la pertenencia de los símbolos a los modelos, es decir, todos los símbolos correspondientes a los centros del diccionario de una palabra pertenecen al modelo de dicha palabra, por lo que la única información suministrada por el modelo de Markov es la relativa al secuenciamiento temporal de dichos símbolos. De ésta forma, aunque dos palabras tengan conjuntos de centros claramente diferentes en sus diccionarios VQ, es posible que el secuenciamiento temporal de éstos sea similar, provocando que las probabilidades de generación de estos dos modelos sean similares, lo que puede provocar errores en la clasificación de secuencias correspondientes a estas dos palabras.

Por ejemplo supongamos las palabras /CERO/ y /UNO/. Los diccionarios VQ correspondientes a estas palabras contendrán centros correspondientes tanto a las vocales como a las consonantes de las palabras. Cuando una secuencia de vectores correspondiente a la palabra /CERO/ se cuantiza con el diccionario VQ de /UNO/, los segmentos de la vocal /O/ de /CERO/ se cuantizarán generando símbolos correspondientes a la vocal /O/ de /UNO/, los correspondientes a la vocal /E/ de /CERO/ generarán probablemente símbolos correspondientes a la vocal /U/ de /UNO/, y por último, los segmentos correspondientes a la consonante /R/ generarán probablemente símbolos correspondientes a la consonante /N/ de /UNO/. Independientemente del tipo de símbolos generados por la consonante /C/ de /CERO/, que son una fracción relativamente pequeña del total de símbolos generados para /CERO/, las secuencias de símbolos generadas por la cuantización de /CERO/ con el diccionario de /UNO/, serán similares a las obtenidas al cuantizar /UNO/ con su propio diccionario; consecuentemente, la discriminación de los modelos de Markov correspondientes a las palabras /CERO/ y /UNO/ será pequeña.

También pueden producirse situaciones a la inversa, en las que sonidos espectralmente similares en posiciones distintas de dos palabras, provoque similares probabilidades de cuantización, pero probabilidades de generación de símbolos claramente diferentes, en estas situaciones es cuando los modelos de Markov realizan la discriminación. Como ejemplo supongamos las palabras /MANO/ y /MONA/. Salvo algunas variantes debidas a las coarticulaciones con los fonemas adyacentes, las consonantes y vocales de las dos palabras tendrán características espectrales muy similares, por lo que las probabilidades de cuantización de cualquiera de ellas con el diccionario con cualquiera de los dos diccionarios serán muy similares. Sin embargo, la precedencia de los símbolos es diferente, así, en el modelo HMM de /MANO/ los símbolos correspondientes a la vocal /A/ aparecerán en estados anteriores a los correspondientes a la vocal /O/, y en el modelo HMM de /MONA/ ocurre lo contrario, por lo que cabe esperar que las probabilidades de generación sean claramente diferentes.

La característica que aporte mayor información para la discriminación dependerá del tipo de vocabulario empleado. Así, con un vocabulario como el utilizado en este trabajo, en el que las palabras están claramente diferenciadas en cuanto a los fonemas constitutivos, una parte importante de la información utilizada en la discriminación será suministrada por las probabilidades de cuantización. Sin embargo, cabe esperar que en un vocabulario formado por palabras con características acústicas similares, los modelos HMM aporten mayor cantidad de información al proceso de clasificación.

El resultado puesto de manifiesto anteriormente referente a que las probabilidades de cuantización, que representan las probabilidades de pertenencia de los diferentes vectores de características a los modelos, aporta la mayor parte de la información utilizada por el sistema en el proceso de clasificación, no es un resultado sorprendente aunque a priori pueda parecerlo. De hecho, una situación similar se observa para el caso de los modelos discretos, si comparamos los errores de reconocimiento en los modelos discretos con 10 estados, con los obtenidos para un único estado por modelo. Al considerar un único estado por modelo se elimina la información sobre el alineamiento temporal de los símbolos, y la única información restante es la correspondiente a las probabilidades de observación de símbolos en el modelo. Esto es equivalente, en cierta forma, a la utilización de las probabilidades de cuantización, en el sentido de que las dos probabilidades se refieren a la probabilidad de observación de un determinado conjunto de vectores dado un modelo. La

diferencia estriba en la forma en que tal probabilidad es estimada, las probabilidades de cuantización utilizan una estimación paramétrica, mientras que las probabilidades de observación de símbolos utilizan una estimación no paramétrica a través de las probabilidades de observación de los símbolos asociados (en el sentido de la cuantización vectorial) a los vectores.

En la tabla X se muestran los errores de reconocimiento para modelos discretos con uno y con diez estados por modelo. Los resultados corresponden a experimentos L1OUT y L4OUT.

Ns	L1OUT		L4OUT	
	1 estado	10 estados	1 estado	10 estados
64	11.67%	3.85%	12.87%	4.37%
128	6.87%	2.61%	9.58%	3.59%
256	4.90%	2.09%	7.45%	2.81%
512	3.75%	1.51%	6.93%	2.03%

Tabla X. Error de reconocimiento para los DHMM con 1 y 10 estados.

De los resultados mostrados en la tabla, puede deducirse que gran parte de la información utilizada en el reconocimiento está presente cuando únicamente se considera un estado por modelo, quedando eliminada consecuentemente la información sobre alineamiento temporal.

Por último, si comparamos los resultados obtenidos para los modelos discretos con un único estado con los correspondientes a los modelos MVQHMM cuando únicamente se utilizan las probabilidades de cuantización para la clasificación de las secuencias de vectores incógnita, vemos que éstos indican que este último método es más eficiente en cuanto a la caracterización de las probabilidades de pertenencia de los vectores a los modelos, dado que se obtienen mejores tasas de reconocimiento incluso cuando únicamente se utilizan 4 centros en cada diccionario VQ de los MVQHMM, en comparación con los DHMM con un número equivalente de centros.

Variación del error con el número de centros por modelo

En la tabla IX (columna P_{TOT}) y en la figura 6.4 se muestra la variación del error de reconocimiento de los modelos MVQHMM frente al número de centros considerado para los diccionarios VQ de los modelos. Los resutados muestran que el rendimiento de los MVQHMM decae rápidamente cuando se reduce el número de centros de los diccionarios VQ de los modelos; ésto es debido a que los centros no son compartidos como en el caso de los DHMM o SCHMM, de forma que las reducciones en el número total de centros afectan más drásticamente al modelado MVQHMM que al modelado DHMM o SCHMM. En cualquier caso, este efecto es sólo perjudicial cuando el número toal de centros considerado es pequeño (p.e. 64 centros en los DHMM o 4 centros por modelo en los MVQHMM). A partir de 8 ó 16 centros por modelo, los resultados son similares a los obtenidos con los DHMM, lo que indica que el modelado realizado por los diccionarios VQ de los modelos MVQHMM es adecuado en comparación con el realizado por un diccionario común VQ con un número total de centros igual a la suma de los centros de los diccionarios VQ de cada uno de los modelos MVQHMM.

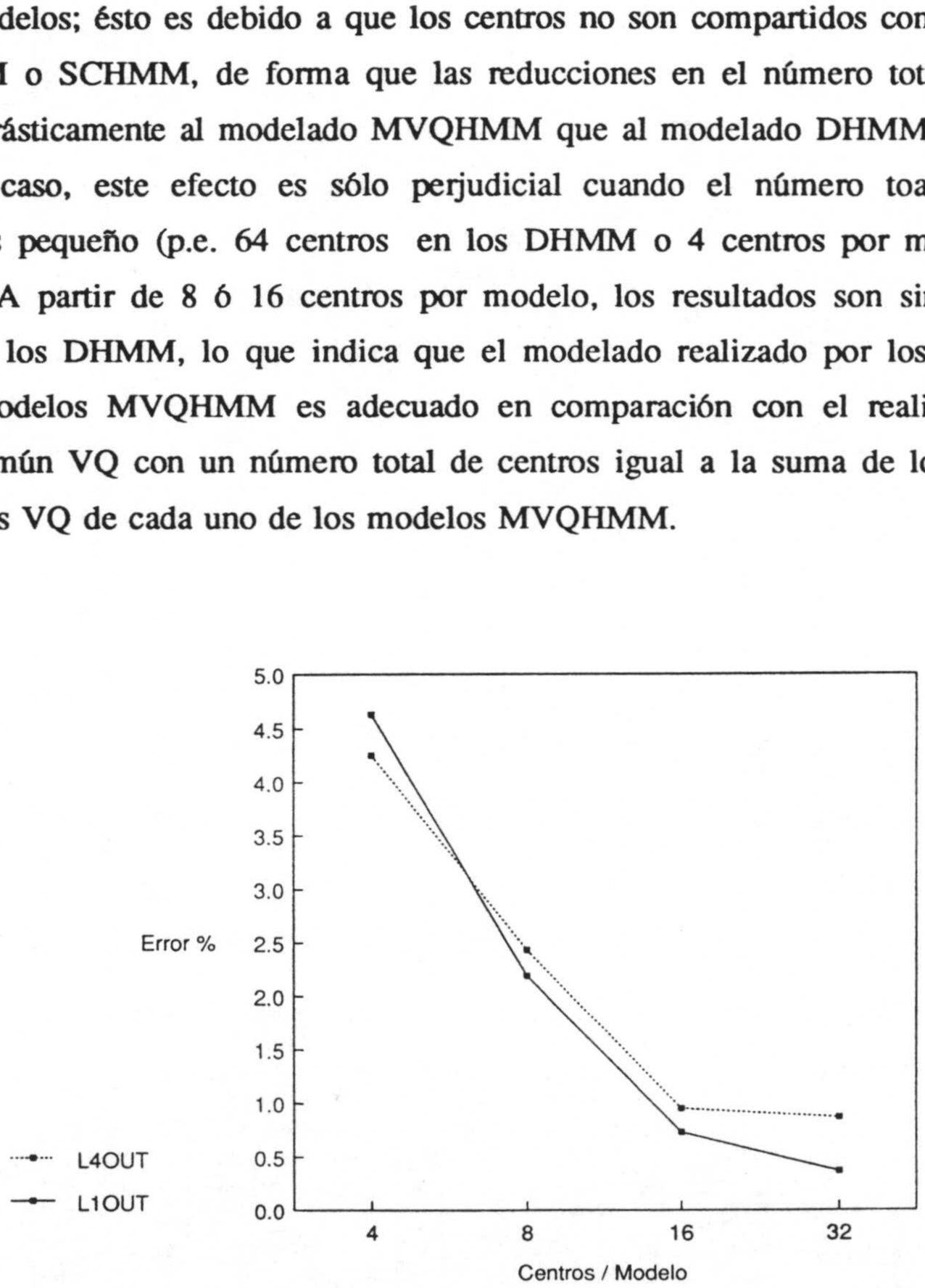

Figura 6.4. Error frente a Ns para los modelos MVQHMM

6.7. COMPARACION DE RESULTADOS

En esta sección presentaremos los resultados comparativos de los tres tipos de modelado utilizados en el presente trabajo. En la tabla XI y la figura 6.5 se muestran los resultados comparativos para los tres tipos de modelados DHMM, SCHMM y MVQHMM, para diferentes configuraciones de número de estados. La columna Nspal-Nstot refleja el número de centros considerado en los modelos MVQHMM, y el número de centros considerados para cada uno de los modelos DHMM y SCHMM. Cada una de las filas de dicha tabla presentan los resultados obtenidos cuando el número total de centros en los tres tipos de modelado es igual (número de centros en el modelado DHMM y SCHMM igual a la suma del número de centros utilizado para cada diccionario VQ de los MVQHMM).

Nspal-Nstot	L1OUT			L4OUT		
	DHMM	SCHMM	MVQHMM	DHMM	SCHMM	MVQHMM
4-64	3.08%	1.77%	4.63%	4.37%	3.02%	4.25%
8-128	1.93%	1.14%	2.19%	3.59%	2.13%	2.43%
16-256	1.36%	0.73%	0.73%	2.81%	1.56%	0.95%
32-512	0.99%	0.52%	0.37%	2.03%	1.43%	0.87%

Tabla XI. Errores de reconocimiento para los modelados DHMM, SCHMM y MVQHMM.

Los resultados muestran que, para la configuración con 4 centros en los diccionarios de los modelos MVQHMM el rendimiento obtenido es claramente inferior que para los modelados DHMM y SCHMM, pero que para 8 centros por modelo, los MVQHMM ofrecen resultados similares a los DHMM en el caso multilocutor, y superiores en el caso independiente del locutor, configuración para la que el error es similar al obtenido para el modelado SCHMM. Cuando el número de centros se aumenta a 16 por modelo, los resultados son claramente superiores a los obtenidos para el modelado DHMM y similares a los obtenidos con el modelado SCHMM para el caso L1OUT, y claramente superiores a los obtenidos en ambos tipos de modelado para la configuración independiente del locutor.

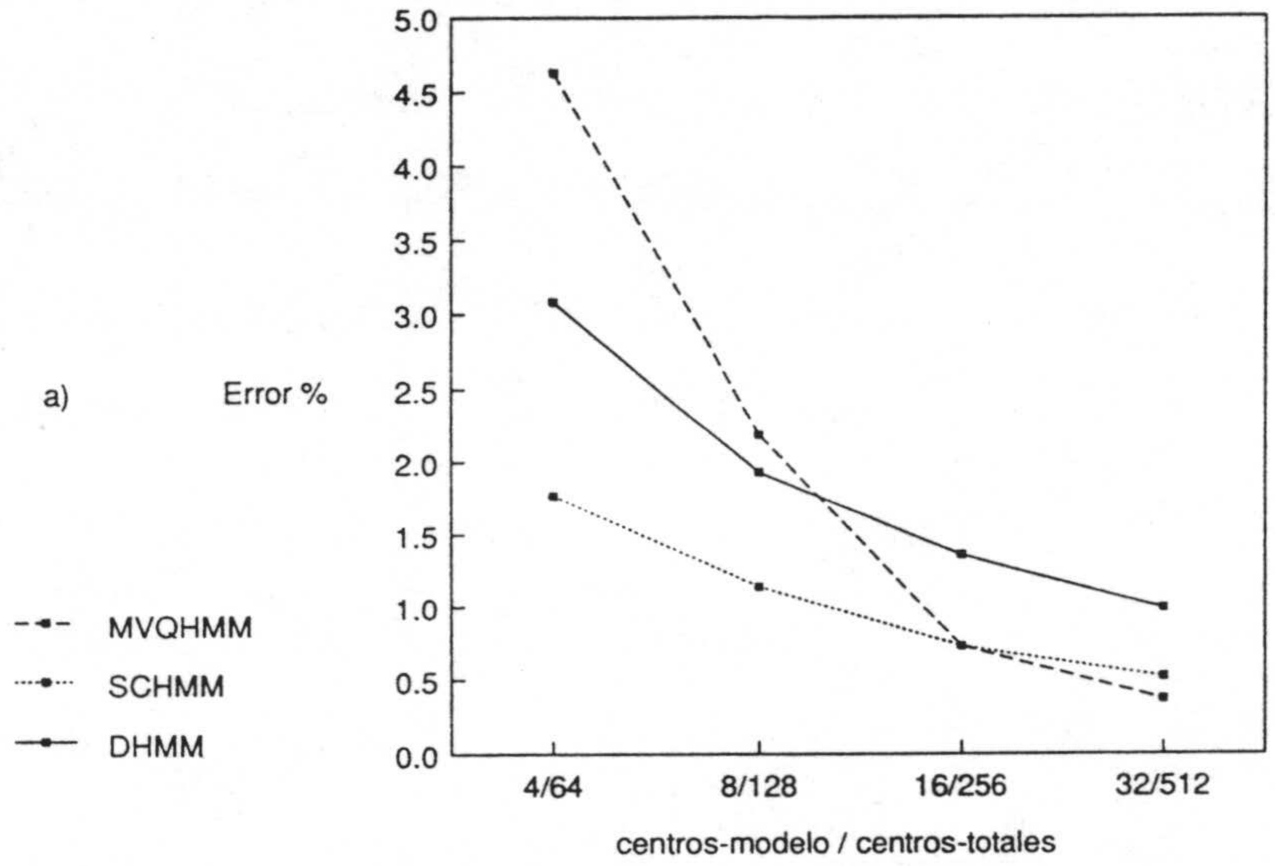

Figura 6.5a. Comparación de resultados configuración L1OUT

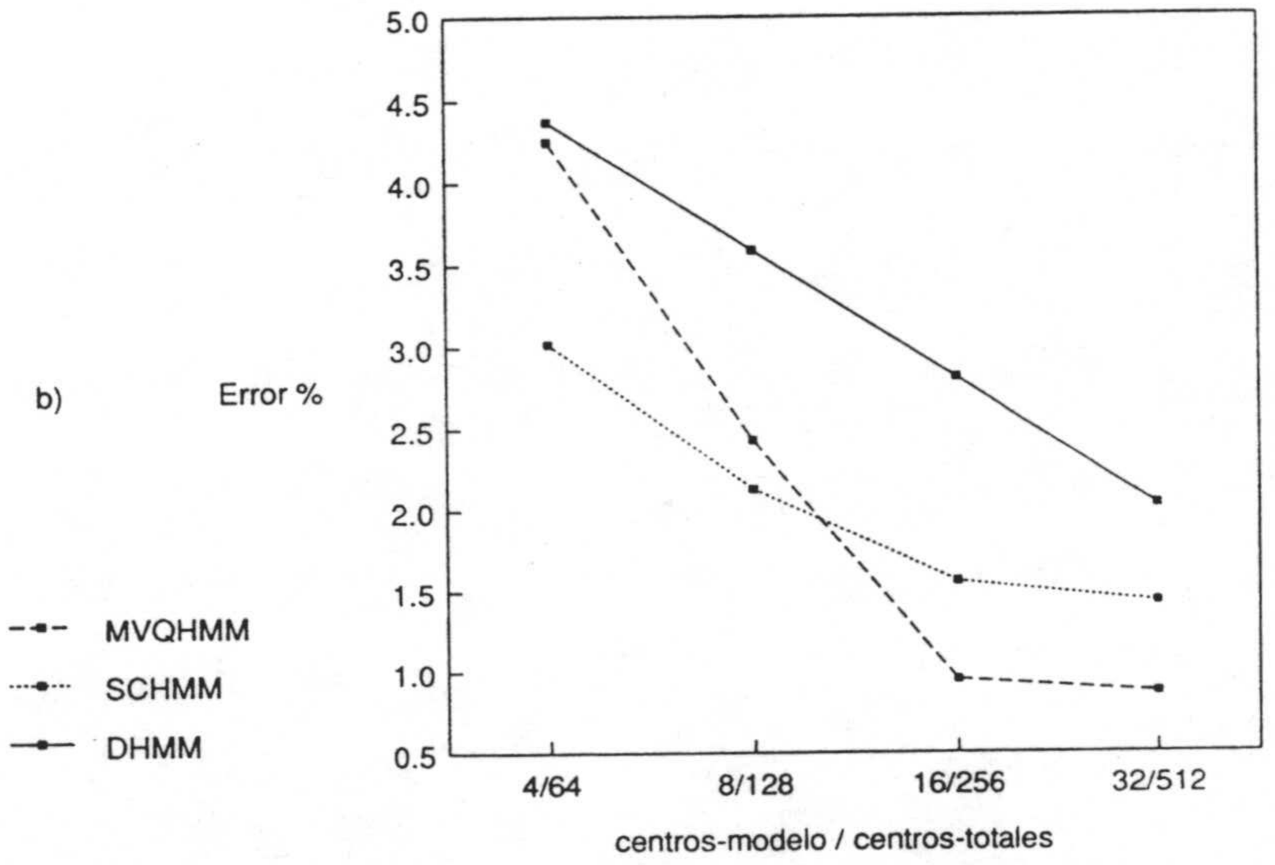

Figura 6.5b. Comparación de resultados configuración L4OUT

La configuración del sistema con 16 centros por modelo (o equivalentemente 256 centros para los DHMM y SCHMM), parece ser la más adecuada para el modelado MVQHMM, obteniéndose sólo pequeños decrementos en el error total de reconocimiento al doblar el número total de centros considerado. Para ésta configuración, los modelos

MVQHMM ofrecen resultados de reconocimiento claramente superiores a los modelos DHMM, y similares (en el caso multilocutor) y superiores (en el caso independiente del locutor) a los modelos SCHMM.

Con respecto a esta última comparación, es de notar que los modelos MVQHMM conllevan una complejidad de cálculo igual a la de los modelos DHMM, y menor que la mitad de la requerida por los modelos SCHMM, para igual número de centros totales, tal y como indicamos en secciones anteriores, por lo que la aplicación del modelado MVQHMM es preferible al modelado SCHMM en cuanto a complejidad computacional.

Una última consideración es la relativa a la comparación de los resultados en las configuraciones multilocutor e independiente del locutor para los tres tipos de modelado. En las figuras 6.4 y 6.6 se muestran los errores de reconocimiento para las configuraciones multilocutor e independiente del locutor de los tres tipos de modelado considerados. Si se comparan las diferencias en el error de reconocimiento de las configuraciones multilocutor e independiente del locutor para los tres métodos de modelado, se comprueba que éstas son mucho menores para el modelado MVQHMM, lo que parece indicar que este tipo de modelado es menos dependiente del conjunto de locutores utilizados en el entrenamiento

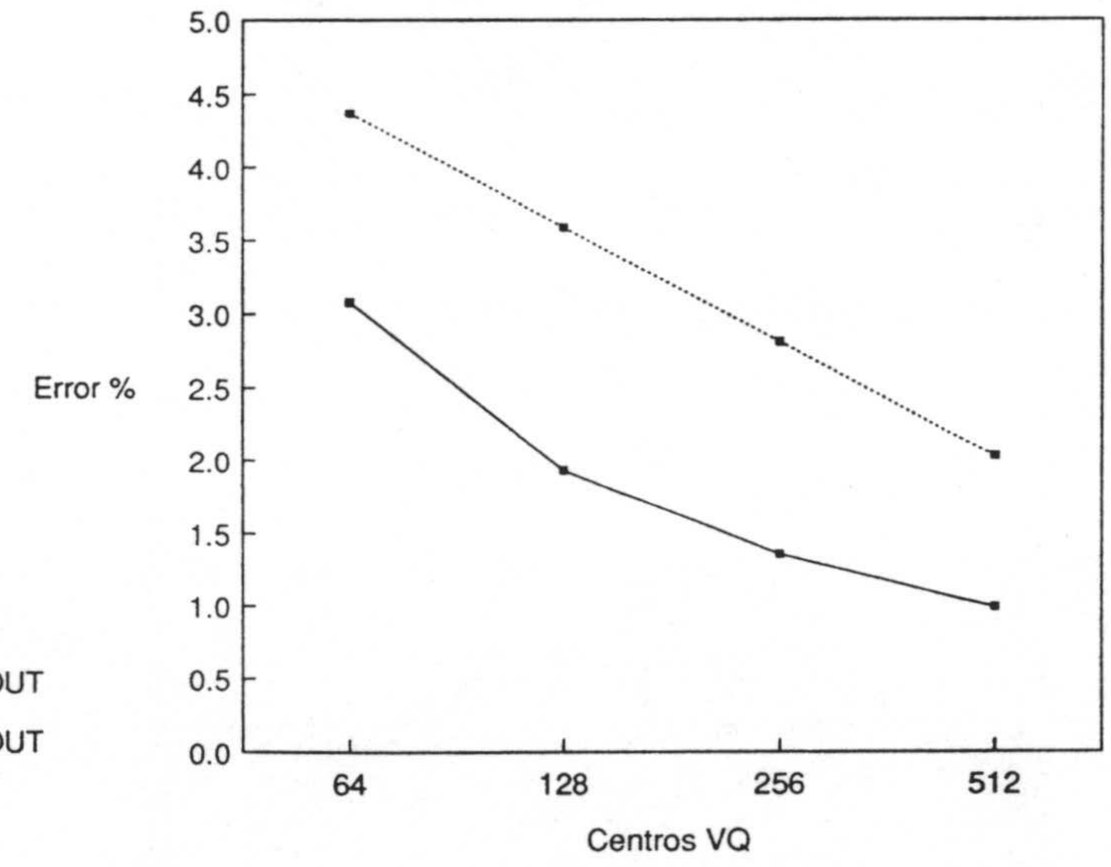

Figura 6.6a. Errores de reconocimiento para el modelado DHMM

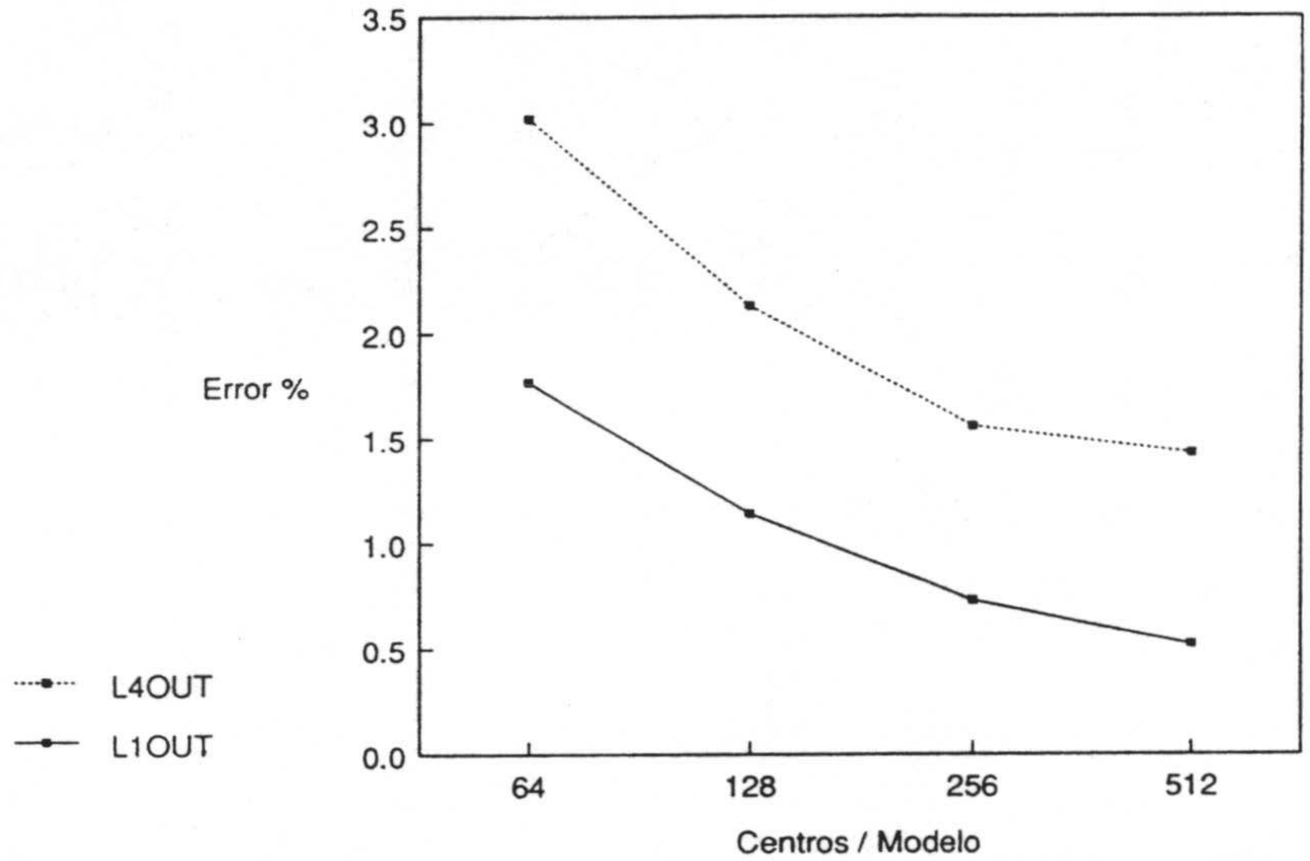

Figura 6.6b. Errores de reconocimiento para el modelado SCHMM

del sistema, ya que no existe mucha diferencia entre los resultados obtenidos por el sistema cuando los locutores del conjunto de test forman parte del conjunto de entrenamiento (configuración multilocutor) y los obtenidos con locutores de test distintos a los empleados en el conjunto de entrenamiento (configuración independiente del locutor). Esta es una característica claramente deseable cuando se pretende diseñar un sistema de reconocimiento independiente del locutor dado que éste necesitará de un número de locutores menor en el conjunto de entrenamiento.

6.8. REDUCCION DEL NUMERO TOTAL DE CENTROS

Aunque los modelos MVQHMM ofrecen resultados similares e incluso superiores a los modelos semicontínuos, con unos requerimientos de cálculo menores, presentan el problema de que su rendimiento decae rápidamente cuando se reduce el número total de centros considerado. El efecto de esta reducción es más drástico en el caso de los modelos MVQHMM que en el caso de los modelos DHMM o SCHMM, debido a que ésta reducción implica una reducción en el número de centros utilizado en los diccionarios individuales de cada uno de los modelos.

En el caso de los modelos DHMM y SCHMM, al compartir un conjunto común de centros de un diccionario universal, la reducción en el número de centros del diccionario no implica necesariamente la reducción del número de centros utilizado por cada uno de los modelos, ya que éstos son compartidos, de forma que un mismo centro puede ser utilizado por varios modelos.

Una forma de evitar este problema es la propuesta por Pan [Pan85] y desarrollada por Furui [Furui88]. Básicamente el método consiste en forzar a que los modelos compartan centros de un diccionario universal, definiendo los diccionarios individuales de los diferentes modelos como subconjuntos de centros de un diccionario común a todos los modelos. De esta forma, se puede conseguir una reducción en el número de centros totales considerado agrupando los centros similares de los diccionarios individuales de los modelos.

6.8.1. CONSTRUCCION DEL DICCIONARIO UNIVERSAL

Una posible aproximación a la construcción del diccionario universal es la de utilizar un conjunto de entrenamiento formado por vectores corespondientes a todas las palabras del vocabulario para la construción de un diccionario independiente de los modelos, y establecer una correspondencia entre los centros de los diccionarios individuales de cada modelo sobre centros de éste diccionario, sustituyendo los centros de los diccionarios individuales por los centros más similares del diccionario universal. Esta aproximación tiene la desventaja de que no existe la certeza de que todos los centros del diccionario universal sean utilizados en los diccionarios individuales de los modelos, con lo que podrían existir centros no utilizados en el diccionario universal.

Una alternativa es la de agrupar los centros de los diccionarios individuales para obtener los centros del diccionario universal. De esta forma existe la certeza de que todos los centros del diccionario universal serán utilizados por los diccionarios individuales de los modelos.

En el presente trabajo, utilizamos un algoritmo K-medias jerárquico similar al utilizado en la construcción de los diccionarios individuales de los modelos, para obtener un agrupamiento de dichos centros. Una vez agrupados los centros de los diccionarios

individuales, cada una de las agrupaciones se sustituye por la media de los centros que la componen, y todos los centros de los diccionarios individuales pertenencientes a un agrupamiento son sustituidos por la media (centro) del mismo. De esta forma, los centros similares de los diccionarios individuales son sustituidos por un valor promedio que corresponde a un centro compartido del diccionario universal. Supuesto que ningún agrupamiento queda vacío en el proceso de construcción del diccionario universal, al menos un centro de un diccionario de algún modelo utiliza cada centro (agrupamiento de centros) del diccionario universal, y no queda ninguno sin utilizar.

En el proceso de cuantización vectorial, dado que los diccionarios de los modelos son subconjuntos de centros del diccionario universal, basta con calcular las distancias de los vectores a los centros del diccionario universal para obtener las cuantizaciones correspondientes a los diferentes diccionarios de los modelos, así como las correspondientes distorsiones de cuantización.

Reducción de centros en los diccionarios individuales

Aunque el propósito del método de reducción de centros es el de agrupar centros similares de palabras diferentes, un efecto secundario del algoritmo utilizado es la reducción del número de centros en los diccionarios individuales. Esto es debido al criterio utilizado para la construcción del diccionario universal, basado en la minimización global de la distancia mínima media entre los centros de los diccionarios individuales y los del diccionario universal.

Durante el proceso de construcción del diccionario universal, es posible que el algoritmo K-medias decida la construcción de un agrupamiento que contiene dos o más centros del diccionario de una palabra, lo que provocará una reducción en el número total de centros de dicho diccionario.

En las tablas XII y XIII se muestra el número medio de centros de los diccionarios individuales resultantes del proceso de reducción para diferentes configuraciones de número de centros por palabra (Nspal) y número total de centros (Nstot).

Nstot / Nspal	64	128	256	512	1024
4	4				
8	7.87	8			
16	13.50	14.94	16		
32	18.56	21.00	29.00	32	
64	22.38	33.25	43.88	55.88	64

Tabla XII. Número medio de centros para la reducción L1OUT

Nstot / Nspal	64	128	256	512	1024
4	4				
8	7.31	8			
16	13.31	14.25	16		
32	19.06	23.63	26.44	32	
64	24.63	32.81	44.44	55.55	64

Tabla XIII. Número medio de centros para la reducción L4OUT

En las tablas de puede observar la reducción del número medio de centros de los diccionarios individuales antes citada, así, para la configuración multilocutor (L1OUT) Nspal=32 Nstot=64, el número medio de centros de los diccionarios individuales se reduce en un factor 1.72 (de 32 a 18.56). Sin embargo, el número total de centros se ha reducido en un factor 8 (de 512 a 64), es decir, que el efecto predominante de la reducción es la compartición de centros entre diferentes diccionarios, tal y como es el propósito del algoritmo. En este ejemplo concreto, el diccionario resultante contiene 64 centros, y los individuales contienen una media de 18.56 centros, con lo que el número medio de

palabras que comparten cada uno de los centros del diccionario universal es de (18.56*16)/64 =4.64.

6.8.2. RESULTADOS DE RECONOCIMIENTO

Para determinar la efectividad del método de agrupamiento de centros antes expuesto, se realizaron experimentos L1OUT y L4OUT para diferentes valores de número de centros en los diccionarios individuales y universal, obteniendo los resultados mostrados en las tablas XIV y XV. Los valores en cada recuadro de las tablas se refieren a $(P_{GEN}+P_{DUR}) + P_{DIST}$ (arriba) y P_{TOT} (abajo).

De los resultados mostrados en las tabla se deduce que la reducción del número de centros de los diccionarios es efectiva cuando el número de centros totales considerado es pequeño, así, para 64 y 128 centros totales, se obtienen configuraciones con 32 ó 64 centros por modelo, con tasas de reconocimiento superiores a las configuraciones correspondientes a 64 ó 128 centros totales sin compartición de centros.

Nstot Nspal	64	128	256	512	1024
4	25.37+7.71 4.63				
8	13.54+10.21 3.54	8.90+4.12 2.19			
16	10.00+14.32 3.44	7.29+5.36 1.87	6.51+1.87 0.73		
32	8.34+16.51 3.28	6.35+7.71 2.24	5.00+3.75 0.88	4.69+1.35 0.37	
64	5.83+17.19 3.07	5.21+8.54 1.62	4.43+4.32 0.99	4.11+2.29 0.83	3.70+1.20 0.42

Tabla XIV. Error de reconocimiento para la reducción de centros L1OUT.

Como se puede observar en las tablas, la reducción del número de centros causa una menor discriminación en las probabilidades de cuantización ya que al existir centros compartidos entre diferentes modelos, las diferencias entre las distorsiones de cuantización se reducen al cuantizar secuencias de vectores de un modelo con el diccionario de otros modelos que comparte centros con el anterior. Sin embargo, se aumenta la discriminación de las probabilidades de generación al existir un mayor número de centros para cada modelo.

Nstot Nspal	64	128	256	512	1024
4	28.56+9.81 4.25				
8	114.63+13.02 5.47	13.63+4.78 2.43			
16	11.87+13.85 3.59	7.55+6.04 2.50	7.90+2.95 0.95		
32	9.69+18.02 3.54	7.66+7.45 2.29	7.03+4.32 1.98	5.90+3.04 0.87	
64	8.59+23.44 3.38	6.72+12.29 2.76	5.37+7.19 2.29	5.73+4.53 1.51	5.99+3.21 1.13

Tabla XV. Error de reconocimiento para la reducción de centros L4OUT.

En cualquier caso, para números de centros totales correspondientes a las configuraciones cuasi-óptimas de los modelos DHMM y SCHMM (256 ó 512), y dado que el número de centros correspondientes a cada modelo antes de la reducción (16 ó 32) es suficiente para modelar adecuadamente los posibles valores de los vectores correspondientes a las palabras, no se obtiene ninguna ventaja de la reducción del número total de centros. Esto es debido al escaso número de palabras que forman el vocabulario, y es de esperar que la reducción sea efectiva con un mayor número de palabras.

En las figuras 6.7a y 6.7b se muestran los resultados obtenidos para las configuraciones óptimas en la reducción de centros de las tablas XII y XIII. Estas

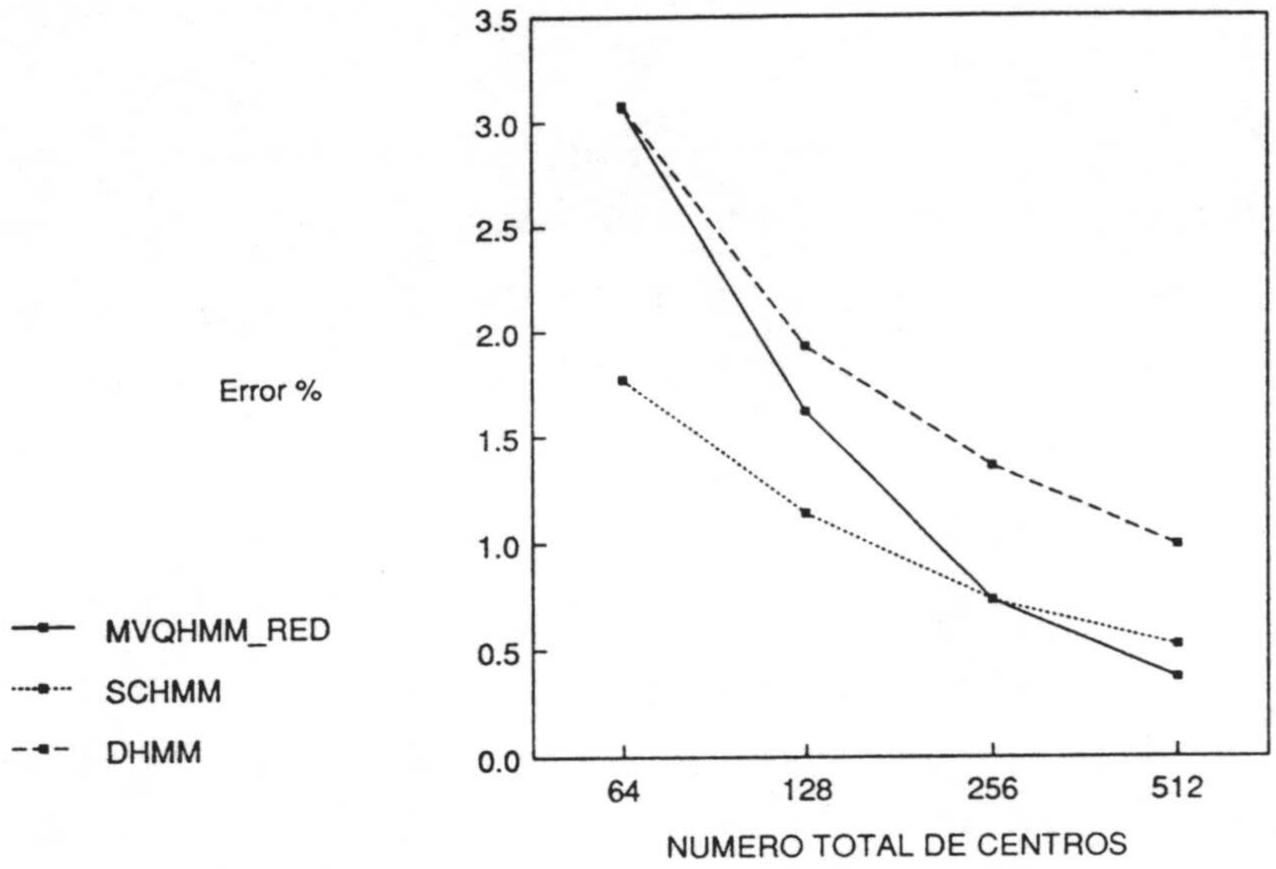

Figura 6.7a. Errores L1OUT para el modelado MVQHMM con reducción de centros

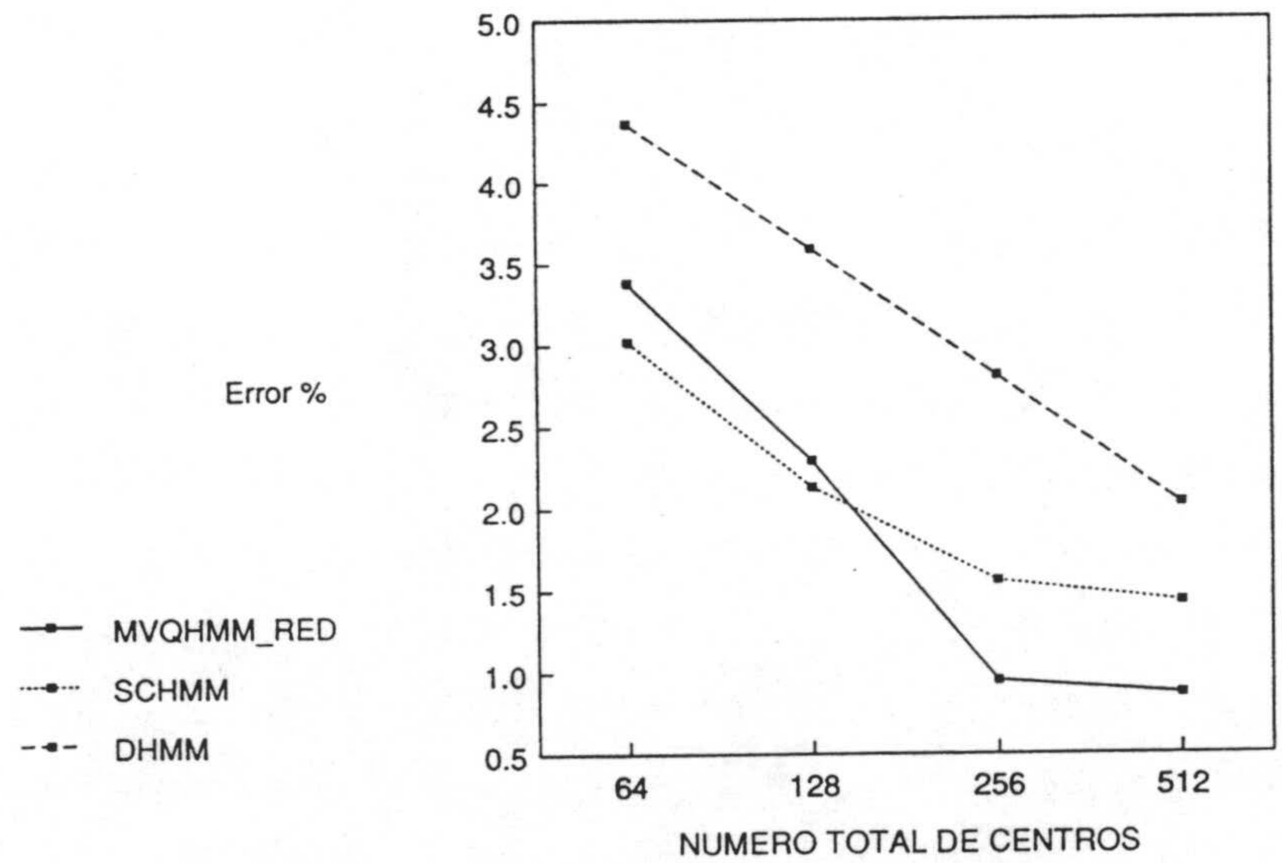

Figura 6.7b. Errores L4OUT para el modelado MVQHMM con reducción de centros

configuraciones son 64/64, 64/128, 16/256 y 32/512 para el caso L1OUT y 64/64, 32/128, 16/256 y 32/512 para el caso L4OUT Los resultados indican que las configuraciones óptimas para los modelos MVQHMM tras la reducción del número de centros totales presentan resultados similares o mejores que los modelos discretos a partir de 64 centros totales, similares a los semicontínuos para un número total de centros de 128, y superiores para 256 y 512 centros.

CAPITULO 7

CONCLUSION

7.1. CONTRIBUCIONES

La principal contribución del presente trabajo es el desarrollo de un sistema de reconocimiento de palabras aisladas, independiente del locutor, con bajo coste computacional y alto rendimiento, con un error de reconocimiento por debajo del 1%.

Partiendo de una configuración básica con modelos ocultos de Markov, se han realizado modificaciones al sistema de reconocimiento para incorporar información adicional al mismo, mejorando su rendimiento. En este sentido, se han mostrado varias posibles modificaciones a la función distancia del cuantizador vectorial, basadas en técnicas de filtrado cepstral del espectro de la señal. También se han añadido características dinámicas del espectro, así como información relativa a las duraciones de los estados de los modelos.

En cuanto a la forma de incorporar las nuevas características (dinámicas) del espectro de la señal al sistema, se han comparado dos aproximaciones diferentes. La primera de ellas basada en la utilización de una distancia compuesta, y la segunda en base a la cuantización independiente de las diferentes características. Los resultados comparativos indican que la segunda de las aproximaciones es sólo ventajosa cuando el número de centros considerados en el diccionarios de cuantización vectorial es reducido, introduciendo en cualquier caso complejidad adicional y mayores requerimientos de memoria y potencia de cálculo en la evaluación de los modelos de Markov.

Se ha implementado un sistema de reconocimiento basado en modelos semicontínuos de Markov, para el que los resultados muestran que la utilización de múltiples candidatos

en el proceso de cuantización vectorial, mejora el rendimiento del sistema, aunque a costa de un incremento en los requerimientos de memoria y tiempo de proceso en la evaluación de los modelos.

Se ha desarrollado una nueva variante del modelado oculto de Markov discreto, los denominados MVQHMM (modelos ocultos de Markov con cuantización dependiente). La principal caracteristicas de esta variante es la utilización de diccionarios VQ específicos para cada uno de los modelos, en lugar de un único diccionario común a todos ellos. De esta forma un modelo MVQHMM está compuesto por un diccionario VQ, que modela los diferentes tipos de espectros correspondientes al modelo, y un HMM discreto, que modela el secuenciamiento temporal de los prototipos espectrales.

Se han deducido las fórmulas de evaluación para los MVQHMM, que permiten la integración natural de la información contenida en las distorsiones de cuantización, y en las probabilidades de generación de los modelos ocultos de Markov, en un contexto probabilístico.

Los estudios comparativos de rendimiento de los sistemas de reconocimiento basados en HMM discretos estandar, HMM semicontínuos y MVQHMM muestran que estos últimos ofrecen tasas de reconocimiento superiores a los dos anteriores, con un coste computacional idéntico al de los modelos HMM discretos estadar. La reducción relativa en el error es del orden del 66% frente a los modelos discretos, y del 39% frente a los semicontínuos.

Se ha mostrado, mediante estudios comparativos de los resultados del sistema de reconocimiento sobre configuraciones *multilocutor* e *independiente del locutor*, que los modelos MVQHMM son más independientes del conjunto de locutores de entrenamiento que los modelos HMM discretos estandar y semicontínuos. Esta caracterítica es deseable cuando se diseña un sistema de reconocimiento independiente del locutor.

Se ha desarrollado un algoritmo para la reducción del número total de centros considerados en los diccionarios VQ de los modelos MVQHMM. Este algoritmo fuerza la compartición de centros entre diccionarios VQ correspondientes a modelos diferentes, consiguiendo una reducción efectiva del número total de centros considerado.

Se ha mostrado que la reducción del número de centros VQ totales deteriora el rendimiento del sistema cuando el número inicial de centros considerados para cada modelo es suficientemente elevado para caracterizar los diferentes tipos de espectros. Sin embargo, cuando el número total de centros considerado es tal que el correspondiente a cada modelo es reducido, la compartición de centros entre diferentes modelos aumenta el rendimiento del sistema.

7.2. TRABAJO FUTURO

Tal y como se indicó en la sección 6.5.1, en el proceso de entrenamiento de los modelos MVQHMM, la optimización de las probabilidades de generación de los modelos requiere la maximización conjunta de las probabilidades de cuantización y las de generación de los modelos de Markov. Sin embargo, en la implementación utilizada en el presente trabajo, el entrenamiento de los modelos se realizó en dos fases separadas e independientes. En la primera de ellas, se optimizan las probabilidades de cuantización mediante la construcción de los diccionarios VQ de los modelos. A continuación, y con las secuencias de símbolos generadas por la cuantización con los diccionarios VQ previamente construidos, se maximizan las probabilidades de generación de los modelos de Markov. Dado que este es un método claramente sub-óptimo, una posible mejora al proceso de entrenamiento del sistema es la de utilizar un método de estimación iterativa como el propuesto por Huang [Huang90]. En un esquema como éste, se pueden reestimar los centros de los diccionarios VQ en base a las probabilidades de generación de los modelos MVQHMM, y una vez modificados los diccionarios VQ, utilizar las nuevas cuantizaciones producidas por éstos para reestimar los modelos de Markov, consiguiendo un nuevo conjunto de modelos MVQHMM (diccionarios VQ y modelos HMM).

Los prometedores resultados ofrecidos por los modelos MVQHMM, en comparación con los modelos HMM discretos y semicontínuos, sugieren que la integración de éstos en sistemas de reconocimiento de voz contínua, como sustitutos de los modelos discretos o semicontínuos, para el modelado acústico de las unidades de decisión (p.e. fonemas, etc.), debe llevar a un incremento en el rendimiento de tales sistemas. Esto es por lo que una segunda tarea a abordar es la del diseño de un sistema de reconocimiento de voz contínua que utilice modelos MVQHMM para el modelado acústico de las unidades de decisión.

APENDICE A

TRANSFORMACION BILINEAL DEL CEPSTRUM

Este apéndice describe la utilización de la transformación bilineal para la conversión en la escala de frecuencias del cepstrum de un señal. Esta conversión se utiliza en el capítulo 5 para la conversión aproximada de los coeficientes cepstrum extraidos de segmentos de señal de voz a escala MEL.

Aqui describiremos las fórmulas de conversión en frecuencia de los coeficientes cepstrum así como la deducción de la matriz de transformación lineal que realiza dicha conversión, estableciendo una relación de recurrencia para la obtención de dicha matriz de transformación.

A.1 TRANSFORMACION BILINEAL DE LA ESCALA DE FRECUENCIAS

La transformación bilineal [Oppenheim75b] es una transformación del plano complejo z en sí mismo que realiza una transformación no lineal de la escala de frecuencias en la función de transferencia de un sistema digital, y ha sido utilizada para obtener transformaciones en las características espectrales de filtros digitales [Constantinides70].

En términos de la variable compleja z, la transformación bilineal puede escribirse en la forma siguiente

$$z_N^{-1} = \left[\frac{z^{-1} - \alpha}{1 - z^{-1}}\right] \quad ; \quad |\alpha| < 1 \tag{A.1a}$$

$$z^{-1} = \left[\frac{z_N^{-1} + \alpha}{1 + z_N^{-1}}\right] \quad ; \quad |\alpha| < 1 \tag{A.1b}$$

donde z_N^{-1} es la nueva variable compleja. La transformación bilineal de la función de transferencia de un sistema digital se reduce a la realización del cambio de variable antes indicado, es decir

$$F_N(z_N^{-1}) = F(z^{-1}) \tag{A.2}$$

donde $F_N()$ es la función de transferencia resultante del cabio en la escala de frecuencias obtenido con la transformación bilineal.

Para obtener la correspondencia entre las frecuencias angulares original w y transformada w_N, basta con escribir la transformación bilineal sobre el círculo unidad en la forma

$$z^{-1} = e^{-jw} \tag{A.3a}$$

$$z_N^{-1} = e^{-jw_N} \tag{A.3b}$$

$$e^{-jw_N} = \frac{e^{-jw} - \alpha}{1 - \alpha e^{-jw}} \tag{A.3c}$$

$$e^{jw_N} = \frac{e^{jw} - \alpha}{1 - \alpha e^{jw}} \tag{A.3d}$$

la tangente de la nueva frecuencia angular w_N puede escribirse en la forma

$$\operatorname{tg} w_N = \left[\frac{\operatorname{sen} w_N}{\cos w_N}\right] \tag{A.4a}$$

$$= \frac{1}{j}\,\frac{e^{jw_N} - e^{-jw_N}}{e^{jw_N} + e^{-jw_N}} \tag{A.4b}$$

y utilizando (A.3c) y (A.3d), y después de algunas transformaciones sencillas se llega a la expresión

$$\operatorname{tg} w_N = \frac{(1 - \alpha^2)\operatorname{sen} w}{-2\alpha + (1 + \alpha^2)\cos w} \tag{A.5}$$

Esta relación indica que las frecuencias w_N y w están relacionadas no-linealmente, de forma que (ver figura 5.8), valores de α mayores que cero realizan una compresión de la zona baja de frecuencias, y una expansión de la zona alta de frecuencias, mientras que con valores negativos de α se consigue el efecto contrario.

A.2 TRANSFORMACION DE LOS COEFICIENTES CEPSTRUM

A continuación mostraremos cómo utilizar la transformación bilineal para obtener la expresión que relaciona los coeficientes cepstrum correspondientes al desarrollo del espectro logarítmico de una señal en escala lineal de frecuencias, con los coeficientes cepstrum del desarrollo correspondiente a este mismo espectro logarítmico pero con la escala de frecuencias transformada bilinealmente.

Dados los coeficientes cepstrum $c(k)$ de una señal, su espectro logarítmico puede expresarse en términos de éstos en la forma

$$F(w) = \sum_{k=-\infty}^{+\infty} c(k)\, e^{-jkw} \tag{A.6}$$

utilizando la transformación bilineal sobre el círculo unidad del plano z, podemos escribir el espectro logarítmico $F_N(w_N)$ en la nueva escala de frecuencias en la forma

$$F_N(w_N) = \sum_{k=-\infty}^{+\infty} c(k) \left[\frac{e^{-jw_N} + \alpha}{1 + \alpha e^{-jw_N}} \right]^k \tag{A.7}$$

de otro lado, el espectro logarítmico $F_N(w_N)$ tiene un desarrollo en términos de los coeficientes cepstrum $\tilde{c}(k)$ en la nueva escala de frecuencias de la forma

$$F_N(w_N) = \sum_{k=-\infty}^{+\infty} \tilde{c}(k)\, e^{-jkw_N} \tag{A.8}$$

e igualando los dos desarrollos obtenemos

$$\sum_{k=-\infty}^{+\infty} \tilde{c}(k)\, e^{-jkw_N} = \sum_{k=-\infty}^{+\infty} c(k) \left[\frac{e^{-jw_N} + \alpha}{1 + \alpha e^{-jw_N}} \right]^k \qquad \text{(A.9a)}$$

En lo que sigue, y por sencillez, eliminaremos el subíndice N, con lo que (A.9a) puede escribirse en la forma

$$\sum_{k=-\infty}^{+\infty} \tilde{c}(k)\, e^{-jkw} = \sum_{k=-\infty}^{+\infty} c(k) \left[\frac{e^{-jw} + \alpha}{1 + \alpha e^{-jw}} \right]^k \qquad \text{(A.9b)}$$

Calculando ahora la transformada inversa de Fourier de los dos miembros de (A.9b), se llega a las expresiones

$$\tilde{c}(n) = \sum_{k=-\infty}^{+\infty} c(k)\, W(n,k) \qquad \text{(A.10a)}$$

$$W(n,k) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \left[\frac{e^{-jw} + \alpha}{1 + \alpha e^{-jw}} \right]^k e^{jnw}\, dw \qquad \text{(A.10b)}$$

donde $W(n,k)$ son los elementos de la matriz de transformación lineal entre los coeficientes cepstrum originales y los transformados, y (A.10a) establece la transformación lineal entre los dos conjutnos de coeficientes cepstrum.

La integral en (A.10a) puede expresarse de forma más sencilla utilizando el cambio de variable $z = e^{jw}$, con lo que la integral angular se transforma en una integral compleja sobre el círculo unidad

$$W(n,k) = \frac{1}{2\pi j} \oint \left[\frac{z^{-1} + \alpha}{1 + \alpha z^{-1}} \right]^k z^{n-1}\, dz \qquad \text{(A.11)}$$

la evaluación de la integral (A.11) se reduce a la suma de los resíduos [Oppenheim75c] contenidos en el círculo unidad del plano compejo z. A continuación discutimos la resolución de la integral para los diferentes valores de k y n.

Caso n > 0

Tal y como se muestra en la figura A.1, para el caso $n > 0$, la distribución de polos y ceros de la función a integrar depende del valor de k. Para el caso $k > 0$, existe un polo múltiple de orden k dentro del círculo unidad con valor $z = -\alpha$, y un cero múltiple de orden k fuera del círculo unidad con valor $z = -1/\alpha$. Para $k < 0$ existe un cero múltiple de orden k dentro del círculo unidad con valor $z = -\alpha$, y un polo múltiple de orden k fuera del círculo unidad con valor $z = -1/\alpha$.

Caso n < 0

En este caso, existe una singularidad adicional de orden (n-1) en el origen. Para evitar esta singularidad múltiple adicional, se puede transformar la integral (A.11) utilizando la expresión alternativa [Oppenheim75c] para la transformada z inversa siguiente

$$\frac{1}{2\pi j}\oint f(z)\,z^{n-1}\,dz = \frac{1}{2\pi j}\oint f(\frac{1}{z})\,z^{-n-1}\,dz \qquad \text{(A.12)}$$

Esta transformación equivale a reflejar los polos y ceros respecto del círculo unidad, con lo que los polos y ceros exteriores son ahora interiores y viceversa. La distribución de polos y ceros es ahora la mostrada en la figura A.1b. Para el caso $k > 0$, existe un cero múltiple de orden k dentro del círculo unidad con valor $z = -\alpha$, y un polo múltiple de orden k fuera del círculo unidad con valor $z = -1/\alpha$. Para $k < 0$ existe un polo múltiple de orden k dentro del círculo unidad con valor $z = -\alpha$, y un cero múltiple de orden k fuera del círculo unidad con valor $z = -1/\alpha$.

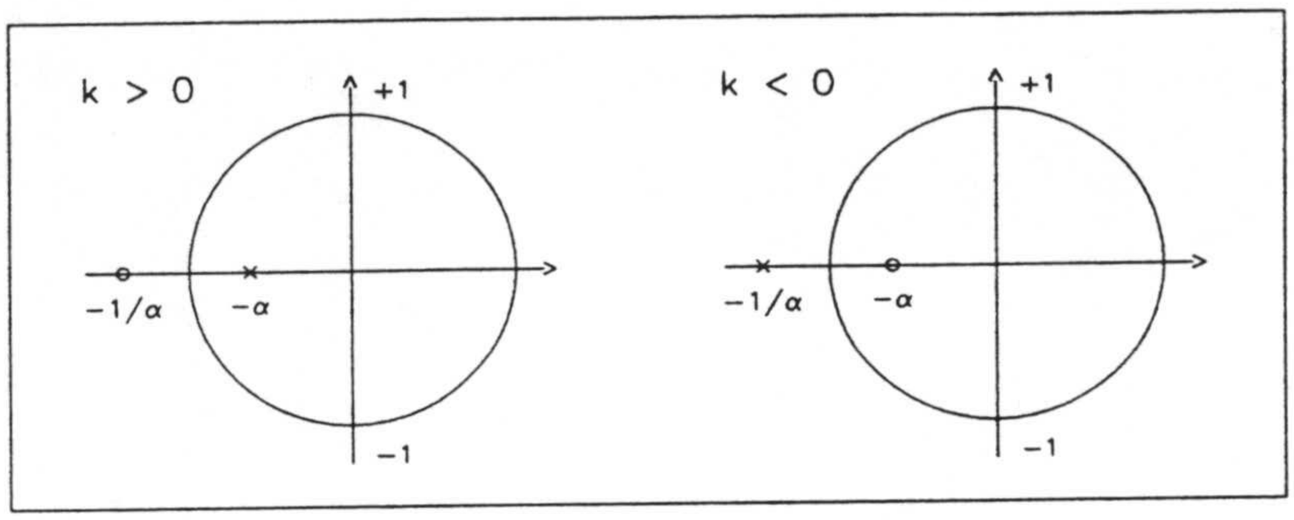

Figura A.1. Polos y ceros para n>0

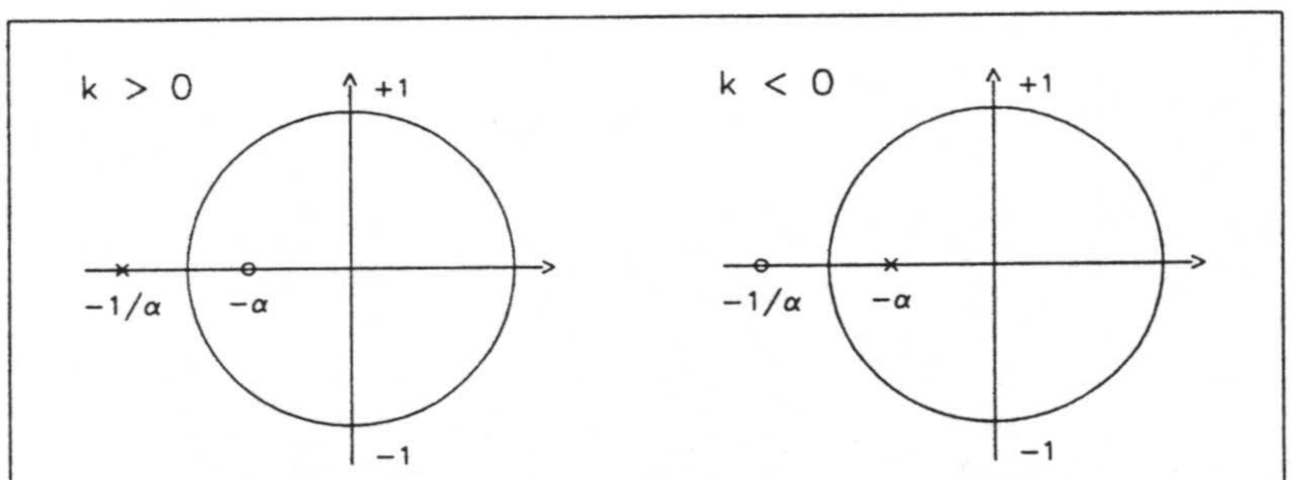

Figura A.2. Polos y ceros para n<0

En resumen, las ecuaciones siguientes corresponden a las cuatro situaciones antes descritas, donde n y k son enteros positivos.

$$W(n,k) = \frac{1}{2\pi j}\oint \left[\frac{z^{-1}+\alpha}{1+\alpha z^{-1}}\right]^k z^{n-1}\, dz \qquad \text{(A.13a)}$$

$$W(n,-k) = \frac{1}{2\pi j}\oint \left[\frac{1+\alpha z^{-1}}{z^{-1}+\alpha}\right]^k z^{n-1}\, dz \qquad \text{(A.13b)}$$

$$W(-n,k) = \frac{1}{2\pi j}\oint \left[\frac{z+\alpha}{1+\alpha z}\right]^k z^{n-1}\, dz \qquad \text{(A.13c)}$$

$$W(-n,-k) = \frac{1}{2\pi j}\oint \left[\frac{1+\alpha z}{z+\alpha}\right]^{k} z^{n-1}\, dz \qquad \text{(A.13d)}$$

Tal y como se deduce de la distribución de polos y ceros de las cuatro integrales, esquematizada en las figuras A.1a y A.1b, los valores de $W(n,-k)$ y $W(-n,k)$ son cero. Además, es sencillo comprobar que las integrales correspondientes a $W(n,k)$ y $W(-n,-k)$ son idénticas dividiendo el numerador y denominador de la fracción en el integrando de (A.13d) por z. Por lo tanto, la expresión para los elementos de W es de la forma

$$W(n,-k) = W(-n,k) = 0 \quad ; \quad n,k>0 \qquad \text{(A.14a)}$$

$$W(n,k) = W(-n,-k) = Res\left\{\left[\frac{1+\alpha z}{z+\alpha}\right]^{k} z^{n-1}\right\}_{z=-\alpha} \quad ; \quad n,k>0 \qquad \text{(A.14b)}$$

donde *Res* denota el resíduo de la expresión.

Caso n = 0

Los polos y ceros para esta situación son los mostrados en la figura A.3. Para el caso $k > 0$ se tiene un polo múltiple de orden k dentro del círculo unidad, más un polo simple en el origen. Para el caso $k < 0$ se tiene únicamente un polo simple en el origen, luego los valores de $W(0,k)$ son

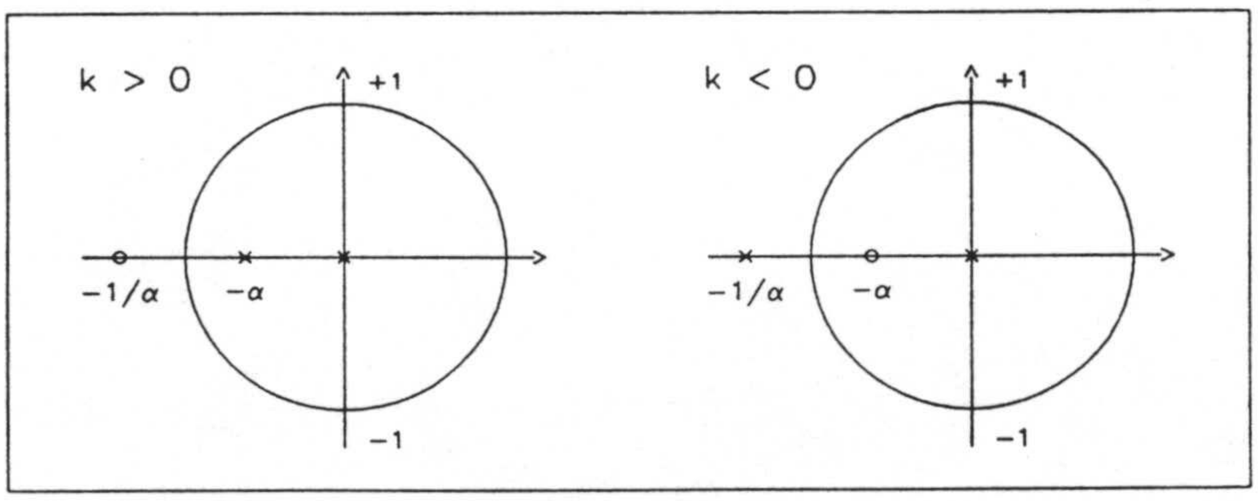

Figura A.3. Polos y ceros para n=0

$$W(0,k) = Res\left\{\left[\frac{1+\alpha z}{z+\alpha}\right]^k z^{-1}\right\}_{z=0} + Res\left\{\left[\frac{1+\alpha z}{z+\alpha}\right]^k z^{-1}\right\}_{z=-\alpha} \quad ; \quad k>0 \text{ (A.15a)}$$

$$W(0,-k) = Res\left\{\left[\frac{z+\alpha}{1+z\,\alpha}\right]^k z^{-1}\right\}_{z=0} \quad ; \quad k<0 \qquad \text{(A.15b)}$$

Por último, queda por resolver la situación para $k = 0$. En esta situación, la integral (A.11) toma la forma

$$W(n,0) = \frac{1}{2\pi j}\oint z^{n-1}\,dz = \delta(n) \quad ; \quad \forall n \qquad \text{(A.16)}$$

Cálculo de W(n,k) y W(-n,-k)

A continuación calcularemos los valores de $W(n,k)$ y $W(-n,-k)$. Como ya indicamos anteriormente, son iguales, y se reducen al cálculo de la expresión A.14b.

$$W(n,k) = Res\left\{\left[\frac{1+\alpha z}{z+\alpha}\right]^k z^{n-1}\right\}_{z=-\alpha} \qquad \text{(A.17a)}$$

$$= \frac{1}{(k-1)!}\left|\frac{d^{k-1}}{dz^{k-1}}(1+\alpha z)z^{n-1}\right|_{z=-\alpha} \qquad \text{(A.17b)}$$

$$= \frac{1}{(k-1)!}\left|\frac{d^{k-1}}{dz^{k-1}}\sum_{l=0}^{k}\binom{k}{l}(\alpha z)^l z^{n-1}\right|_{z=-\alpha} \qquad \text{(A.17c)}$$

$$= \frac{1}{(k-1)!}\sum_{l=0}^{k}\binom{k}{l}\alpha^l\left|\frac{d^{k-1}}{dz^{k-1}}z^{(n+l-1)}\right|_{z=-\alpha} \qquad \text{(A.17d)}$$

$$= \frac{1}{(k-1)!}\sum_{\substack{l=0\\ l\geq(k-n)}}^{k}\binom{k}{l}\alpha^l\frac{(n+l-1)!}{((n+l-1)-(k-1))!}(-\alpha)^{(n+l-1)-(k-1)} \qquad \text{(A.17e)}$$

$$= \sum_{\substack{l=0 \\ l\geq(k-n)}}^{k} \binom{k}{l} \alpha^l \binom{n+l-1}{k-1} (-\alpha)^{(n-k)+l} \tag{A.17f}$$

$$= (-\alpha)^{(n-k)} \sum_{\substack{l=0 \\ l\geq(k-n)}}^{k} \binom{k}{l} \binom{n+l-1}{k-1} (-\alpha^2)^l \tag{A.17g}$$

Donde la condición $l \geq (k-n)$ está impuesta por el hecho de que las derivadas se anulan para valores inferiores de l, dado que los exponentes $(n+l-1)$ son siempre mayores o iguales que cero ya que $n > 0$.

Cálculo de W(0,k) y W(0,-k)

A continuación, calcularemos los valores para $W(0,k)$ y $W(0,-k)$ de forma análoga a como se calcularon $W(n,k)$ y $W(n,-k)$ en la sección precedente. Para ello calcularemos los tres resíduos que figuran en las expresiones (A.15a) y (A.15b).

$$Res\left\{\left[\frac{1+\alpha z}{z+\alpha}\right]^k z^{-1}\right\}_{z=0} = \alpha^{-k} \tag{A.18}$$

$$Res\left\{\left[\frac{1+\alpha z}{z+\alpha}\right]^k z^{-1}\right\}_{z=-\alpha} = \tag{A.19a}$$

$$= \frac{1}{(k-1)!} \left| \frac{d^{k-1}}{dz^{k-1}} (1+\alpha z)^k z^{-1} \right|_{z=-\alpha} \tag{A.19b}$$

$$= \frac{1}{(k-1)!} \left| \frac{d^{k-1}}{dz^{k-1}} \sum_{l=0}^{k} \binom{k}{l} (\alpha z)^k z^{-1} \right|_{z=-\alpha} \tag{A.19c}$$

$$= \frac{1}{(k-1)!} \left| \frac{d^{k-1}}{dz^{k-1}} \sum_{l=0}^{k} \binom{k}{l} \alpha^k \left| \frac{d^{k-1}}{dz^{k-1}} z^{l-1} \right|_{z=-\alpha} \right. \tag{A.19d}$$

a esta sumatoria sólo contribuyen los términos para $l=0$ y $l=k$, dado que los demás se anulan, luego se tiene

$$= \frac{1}{(k-1)!}\binom{k}{0}\alpha^0 \left|\frac{d^{k-1}}{dz^{k-1}} z^{-1}\right|_{z=-\alpha} \qquad \text{(A.19e)}$$

$$+ \frac{1}{(k-1)!}\binom{k}{k}\alpha^k \left|\frac{d^{k-1}}{dz^{k-1}} z^{k-1}\right|_{z=-\alpha}$$

$$= \frac{1}{(k-1)!}\left\{(-1)^{k-1}(k-1)!\,(-\alpha)^{-k} + (k-1)!\,\alpha^k\right\} = \alpha^k - \alpha^{-k} \qquad \text{(A.19f)}$$

$$Res\left\{\left[\frac{z+\alpha}{1+\alpha z}\right]^k z^{-1}\right\}_{z=0} = \alpha^k \qquad \text{(A.20)}$$

Una vez calculados los resíduos necesarios, se pueden calcular los valores de $W(0,k)$ y $W(0,-k)$. Sustituyendo (A.18) y (A.19f) en (A.15) y (A.20) en (A.15b) obtenemos

$$W(0,k) = W(0,-k) = \alpha^k \qquad \text{(A.21)}$$

Relaciones para el cálculo de W(n,k)

Las expresiones siguientes resumen el cálculo de los elementos de la matriz de transformación lineal de los coeficientes cepstrum

$$W(n,k) = W(-n,-k) = (-\alpha)^{(n-k)} \sum_{\substack{l=0 \\ l\geq(k-n)}}^{k} \binom{k}{l}\binom{n+l-1}{k-1}(-\alpha^2)^l \quad ; \quad n,k>0 \qquad \text{(A.22a)}$$

$$W(0,k) = W(0,-k) = \alpha^k \quad ; \quad k>0 \qquad \text{(A.22b)}$$

$$W(n,0) = \delta(n) \quad ; \quad \forall n \qquad \text{(A.22c)}$$

$$W(n,-k) = W(-n,k) = 0 \quad ; \quad n>0\,,\ k>0 \qquad \text{(A.22d)}$$

Una expresión alternativa a (A.21a) puede obtenerse con un desarrollo análogo realizando el cambio de variable

$$(1 + \alpha z) = x$$

en (A.17b) con lo que se obtiene la expresión

$$W(n,k) = (-1)^{(n-1)}\,\alpha^{(k-n)}(1-\alpha^2)\sum_{l=0}^{n-1}\binom{n-1}{l}\binom{k+l}{k-1}(-1)^l\,(1-\alpha^2)^l \quad ; \quad n,k>0 \qquad \text{(A.22e)}$$

Recursión para el cálculo de W(n,k)

La expresión (A.22a) deducida para el cálculo de $W(n,k)$, es compleja debido a la necesidad de la evaluación de los números combinatorios. A continuación obtendremos una expresión recursiva que permite un cálculo más eficiente.

La derivación de la recursión para el cálculo de $W(n,k)$ para $n,k>0$ se puede obtener a partir de la expresión diferencial del resíduo en la forma

$$W(n,k) = Res\left\{\left[\frac{1+\alpha z}{z+\alpha}\right]^k z^{n-1}\right\}_{z=-\alpha} = \frac{1}{(k-1)!}\left|\frac{d^{k-1}}{dz^{k-1}}(1+\alpha z)^k z^{n-1}\right|_{z=-\alpha} \qquad \text{(A.23)}$$

Calculando la primera derivada se obtiene la expresión

$$W(n,k) = \frac{1}{(k-1)!}\left|\frac{d^{k-2}}{dz^{k-2}}\left[k\,\alpha(1+\alpha z)^{k-1}z^{n-1} + (n-1)(1+\alpha z)^k z^{n-2}\right]\right|_{z=-\alpha} \qquad \text{(A.24)}$$

Descomponiendo ahora el término $(1+\alpha z)^k$ en $(1+\alpha z)(1+\alpha z)^{k-1}$ y desarrollando la expresión resultante se obtiene

$$W(n,k) = \frac{1}{(k-1)!} \left| \frac{d^{k-2}}{dz^{k-2}} \left[k\,\alpha(1+\alpha z)^{k-1} z^{n-1} \right] \right|_{z=-\alpha} \qquad \text{(A.25)}$$

$$+ \frac{1}{(k-1)!} \left| \frac{d^{k-2}}{dz^{k-2}} \left[(n-1)(1+\alpha z)^{k-1} z^{n-2} \right] \right|_{z=-\alpha}$$

$$+ \frac{1}{(k-1)!} \left| \frac{d^{k-2}}{dz^{k-2}} \left[\alpha(n-1)(1+\alpha z)^{k-1} z^{n-1} \right] \right|_{z=-\alpha}$$

Si comparamos estas expresiones con (A.23), vemos que, después de algunas modificaciones, se pueden escribir en términos de $W(n,k)$ en la forma siguiente

$$W(n,k) = \frac{(k-2)!}{(k-1)!} \left[k\,\alpha W(n,k-1) + (n-1) W(n-1,k-1) + \alpha(n-1) W(n,k-1) \right] \qquad \text{(A.26)}$$

y agrupando términos y simplificando llegamos a la expresión recursiva

$$W(n,k) = \frac{1}{(k-1)} \left[(n+k-1)\,\alpha W(n,k-1) + (n-1) W(n-1,k-1) \right] ; \quad n,k>1 \qquad \text{(A.27)}$$

Nótese que la relación es válida sólo para $n,k>1$, ya que hemos descompuesto la derivada de orden $k-1$ en una de orden $k-2$ y otra de orden 1, lo que sólo puede hacerse bajo esta suposición. Por lo tanto, es necesario calcular los valores de $W(n,k)$ para los casos $n=0$ y $k=0$.

El valor para $n=0$ ya se calculó anteriormente, y está dado por la expresión (A.22b). En cuanto a los valores para $n=1$, se pueden obtener fácilmente de la resolución directa de (A.13a) ó (A.13d) en la forma siguiente

$$W(1,k) = \frac{1}{2\pi j}\oint\left[\frac{1+\alpha z}{z+\alpha}\right]^k dz \tag{A.28a}$$

$$= Res\left\{\left[\frac{1+\alpha z}{z+\alpha}\right]^k\right\}_{z=-\alpha} \tag{A.28b}$$

$$= \frac{1}{(k-1)!}\left|\frac{d^{k-1}}{dz^{k-1}}(1+\alpha z)^k\right|_{z=-\alpha} \tag{A.28c}$$

$$= \frac{k!}{(k-1)!}\alpha^{k-1}\,|1+\alpha z\,|_{z=-\alpha} \tag{A.28d}$$

$$= k\,\alpha^{k-1}(1-\alpha^2) \quad ; \quad \forall k \geq 2 \tag{A.28e}$$

además es fácil comprobar que esta expresión también valida para los valores k=0,1.

La solución para $k = 0$ también se calculó previamente y está dada por (A.22c). Por último, sólo queda por calcular el caso $k = 1$, cuya solución se puede obtener de forma similar al caso $n = 1$.

$$W(n,1) = \frac{1}{2\pi j}\oint\left[\frac{1+\alpha z}{z+\alpha}\right] z^{n-1} dz \tag{A.29a}$$

$$= Res\left\{\left[\frac{1+\alpha z}{z+\alpha}\right] z^{n-1}\right\}_{z=-\alpha} \tag{A.29b}$$

$$= (-\alpha)^{n-1}(1-\alpha^2) \tag{A.29c}$$

Una vez obtenidos todos los resultados correspondientes a los diferentes casos, la recursión para el cálculo de $W(n,k)$ queda en la forma siguiente

$$W(0,k) = \alpha^k \quad ; \quad n=0\ ,\ k \geq 0 \tag{A.30a}$$

$$W(1,k) = k\,\alpha^{k-1}(1-\alpha^2) \quad ; \quad n=1\ ,\ k \geq 0 \tag{A.30b}$$

$$W(n,0) = 0 \quad ; \quad n \geq 2 \tag{A.30c}$$

$$W(n,1) = (-\alpha)^{n-1}(1-\alpha^2) \quad ; \quad n \geq 2 \tag{A.30d}$$

$$W(n,k) = \frac{1}{(k-1)}\left[(n+k-1)\,\alpha\,W(n,k-1) + (n-1)\,W(n-1,k-1)\right] \ ;\ n \geq 2\ ,\ k \geq 2 \tag{A.30e}$$

BIBLIOGRAFIA

[Abramson74] N. Abramson, *Teoría de la Información y Codificación,* Paraninfo (1974).

[Andreu90] G. Andreu, E. Vidal, and F. Casacuberta, ''An empirical evaluation of feature maps and other clustering techniques for frame labeling of speech,'' *Proc. EUSIPCO'90* **Vol.2** pp. 1267-1270 (1990).

[Appelbaum89] T.H. Appelbaum and B.A. Hanson, ''Enhacing the discrimination of speaker independent hidden Markov models with corrective training,'' *Proc. ICASSP'89* **Vol.1** pp. 302-305 (1989).

[Appelbaum90] T.H. Appelbaum and B.A. Manson, ''Robust speaker-independent word recognition using spectral smoothing and temporal derivatives,'' *Proc. EUSIPCO'90* **Vol.2** pp. 1183-1186 (1990).

[Atal74] B.S. Atal, ''Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification,'' *J. Acoust. Soc. Amer.* **Vol.55** pp. 1304-1312 (June 1974).

[Baker75] J.K. Baker, ''The DRAGON system - An overview,'' *IEEE Trans. Acoust. Speech, Signal Processing* **ASSP-23, No.1** pp. 24-29 (February 1975).

[Bakis76] R. Bakis, "Continuous speech recognition via centisecond acoustic states," *91st Meeting of the Acoustical Society of America*, (April 1976).

[Baum67] L.E. Baum and A. Egon, "An inequality with applications to stochastical estimation for probabilistic funtions of a Markov process and to a model for Ecology," *Bull. Amer. Meteorol. Soc.* **Vol.73** pp. 360-363 (1967).

[Baum68] L.E. Baum and G.R. Sell, "Growth functions for transformations on mainfolds," *Pac. J. Math.* **Vol.27, No.2** pp. 211-227 (1968).

[Bellman57] R.E. Bellman, *Dynamic Programming*, Princenton University Press (1957).

[Bergh85] A.F. Bergh, F.K. Soong, and L.R. Rabiner, "Incorporation of temporal structure into a vector-quantization-based preprocessor for speaker-independent, isolated word recognition," *AT&T Tech. J.* **Vol.64 No.5** pp. 1047-1063 (May-June 1985).

[Bocchieri86] L. Bocchieri and R. Doddington, "Frame specific statistical features for speaker independent speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing* **ASSP-4 No.4** pp. 755-764 (August 1986).

[Bourlard89] H. Bourlard and C.J. Wellekens, "Speech Dynamics and Recurrent Neural Networks," *ICASSP'89*, pp. 33-36 (1989).

[Bridle82] J.S. Bridle, M.D. Brown, and R.M. Chamberlain, "An algorithm for Speech Recognition," *ICASSP'82*, pp. 899-902 (May 1982).

[Bridle90] J.S. Bridle, "Alpha-Nets: A Recurrent 'Neural' Network Architecture with a Hidden Markov Model Interpretation," *Speech Communications* **Vol.9** pp. 83-92 (1990).

[Brown82] M.K. Brown and L.R. Rabiner, ''On the use of energy contours in LPC-based recognition of isolated words,'' *Bell Syst. Tech. J.* **Vol.61 No.10** pp. 2971-2987 (December 1982).

[Brown87] P. Brown, ''The acoustic-modeling problem in automatic speech recognition,'' *PhD Thesis*, Computer Science Department, Carnegie Mellon University, (May 1987).

[Burton85] D.K. Burton, ''Applying Matrix Quantization to Isolated Word Recognition,'' *ICASSP'85*, pp. 29-32 (Tampa 1985).

[Burton85b] D.K. Burton, J.E. Shore, and J.T. Buck, ''Isolated-word recognition using multisection VQ codebooks,'' *IEEE Trans. Acoust., Speech, Signal Processing* **ASSP-33 No.4** pp. 837-849 (August 1985).

[Bush87] M.A. Bush and G.E. Kopec, ''Network-based connected digit recognition,'' *IEEE Trans. Acoust., Speech, Signal Processing* **ASSP-35 No.10** pp. 1401-1413 (October 1987).

[Buzo80] A. Buzo, A.H. Gray, R.M. Gray, and J.D. Markel, ''Speech Coding Based Upon Vector Quantization,'' *IEEE Trans. Acoust. Speech, Signal Processing* **ASSP-28, No.5** pp. 562-574 (October 1980).

[Buzo82] A. Buzo, C. Rivera, and H. Martínez, ''Discrete utterance recognition based upon source coding techniques,'' *ICASSP'82*, pp. 539-542 (1982).

[Casacuberta90] F. Casacuberta, E. Vidal, B. Más, and H. Rulot, ''Learning the structure of HMM's through grammatical inference techniques,'' *Proc. ICASSP'90* **Vol.2** pp. 717-720 (1990).

[Chang90] L. Chang and M.M. Bayoumi, ''Parametric modeling of state transitions in hidden Markov models,'' *Proc. EUSIPCO'90* **Vol.2** pp.

1387-1390 (1990).

[Chow90] Y.L. Chow, "Maximum mutual information estimation of HMM parameters for continuous speech recognition using the n-best algorithm," *Proc. ICASSP'90* **Vol.2** pp. 701-704 (1990).

[Chown87] Y.L Chown, M.O. Dunham, O.A. Kimball, M.A. Krasner, G.F. Kubala, J. Makhoul, S. Roucos, and R.M. Schwartz, "BYBLOS: The BBN Continuous Speech Recognition System," *IEEE Int. Conf. Acoust. Speech, Signal Proc.*, (April 1987).

[Constantinides70] A.G. Constantinides, "Spectral transformation for Digital FIlters," *Proc. IEE* **Vol.117 No.8** pp. 1585-1590 (August 1970).

[Díaz91] J.E. Díaz, A. Peinado, J.C. Segura, M.C. Benítez, and A. Rubio, "Recurrent Neural Networks for Speech Recognition," *Artificial Neural Networks. Proc. IWANN'91*, pp. 361-369 Springer-Verlag, (1991).

[Davis80] S.B. Davis and P. Mermelstein, "Comparison of parametric representations of monosyllabic word recognition in continuous spoken sentences," *IEEE Trans. Acoust. Speech, Signal Processing* **ASSP-24, no.8** pp. 357-366 (August 1980).

[Dempster77] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data and the EM algorithm," *J. Roy. Stat. Soc.* **Vol.39, No.1** pp. 1-38 (1977).

[Duda73c] R.O. Duda and P.E. Hart, "Clustering and Dimensionality Reduction," *Pattern Classification and Scene Analysis* **Vol.I** pp. 243-252 John Wiley & Sons, N.Y., London, Sydney, Toronto, (1973).

[Duda73] R.O. Duda and P.E. Hart, "Estimating the error rate," *Pattern Classification and Scene Analysis* **Vol.I** pp. 211-256 John Wiley & Sons, N.Y., London, Sydney, Toronto, (1973).

[Duda73b] R.O. Duda and P.E. Hart, "Data description and clustering," *Pattern Classification and Scene Analysis* **Vol.I** pp. 211-256 John Wiley & Sons, N.Y., London, Sydney, Toronto, (1973).

[Falaschi90] A. Falaschi and P. Pierucci, "Some experiments on HMM structure inference," *Proc. EUSIPCO'90* **Vol.2** pp. 1375-1378 (1990).

[Fissore90] L. Fissore, M. Cadagno, and G. Pirani, "Isolated word recognition in the mobile-radio system: experiments and results," *Proc. EUSIPCO'90* **Vol.2** pp. 1207-1210 (1990).

[Forney73] G.D. Forney, "The Viterbi algorithm," *Prov. IEEE* **Vol.61** pp. 268-278 (March 1973).

[Furui81] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Trans. Acoust. Speech, Signal Processing* **ASSP-29** pp. 254-272 (1981).

[Furui86] S. Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum," *IEEE Trans. Acoust., Speech, Signal Processing* **ASSP-34 No.1** pp. 52-59 (February 1986).

[Furui88] S. Furui, "A VQ-based preprocessor using cepstral dynamic features for speaker-independent large vocabulary word recognition," *IEEE Trans. Acoust., Speech, Signal Processing* **ASSP-7 No.7** pp. 980-987 (July 1988).

[Gauvani86] J.L Gauvani, "A Syllabe Based Isolated Word Recognition Experiment," *ICASSP'86*, pp. 57-60 (Tokyo 1986).

[Gray76] A. Gray and J. Markel, "Distance Measures for Speech Processing," *IEEE Trans. Acoust. Speech, Signal Processing* **ASSP-24** pp. 380-391 (October 1976).

[Gray80] R.M. Gray, A. Buzo, A.H. Gray, and Y. Matsuyama, "Distortion measures for speech processing," *IEEE Trans. Acoust. Speech, Signal Processing* **ASSP-28, no.4** pp. 367-376 (August 1980).

[Gray84] R.M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, pp. 4-29 (April 1984).

[Gu90] H. Gu, C. Iseng, and L. Lee, "Isolated-utterances speech recognition using hidden Markov models with bounded states duration," *Proc. EUSIPCO'90* **Vol.2** pp. 1283-1286 (1990).

[Gupta87] V.N. Gupta, M. Lennig, and P. Mermelstein, "Integration of acoustic information in a large vocabulary word recognizer," *IEEE Int. Conf. Acoust. Speech, Signal Proc.*, pp. 697-700 (April 1987).

[Huang90b] E.F. Huang and F.K. Soong, "A probabilistic acoustic map based discriminative HMM training," *Proc. ICASSP'90* **Vol.2** pp. 693-696 (1990).

[Huang89b] X.D. Huang and M.A. Jack, "Unified techniques for vector quantisation and hidden Markov modeling using semi-continuous models," *Proc. ICASSP'89*, pp. 639-642 (1989).

[Huang89c] X.D. Huang and M.A. Jack, "Semi-continuous hidden Markov models for speech recognition," *Computer Speech and Language* **Vol.3** pp. 239-251 (1989).

[Huang89] X.D. Huang, H.W. Hon, and K.F. Lee, "Large-vocabulary speaker-independent continuous speech recognition with semi-continuous

hidden Markov models,'' *EUROSPEECH-89*, pp. 163-166 (1989).

[Huang90] X.D. Huang, K.F. Lee, and H.W. Hon, ''On semi-continuous hidden Markov modeling,'' *Proc. ICASSP'90* **Vol.2** pp. 689-692 (1990).

[Itakura75] F. Itakura, ''Minimum prediction residual principle applied to speech recognition,'' *IEEE Trans. Acoust. Speech, Signal Processing* **ASSP-23** pp. 67-72 (February 1975).

[Jelinek76] F. Jelinek, ''Continuous Speech Recognition by statistical methods,'' *IEEE Proc.* **64, No.4** pp. 532-556 (April 1976).

[Jelinek80] F. Jelinek and R.L. Mercer, ''Interpolated estimation of Markov source parameters from sparse data,'' *Pattern Recognition in Practice*, pp. 381-397 E.S. Gelesma and L.N. Kanal Eds., Amsterdam, The Netherlands: North-Holland, (1980).

[Juang85] B.H. Juang and L.R. Rabiner, ''Mixture autoregressive Hidden Markov Models for speech signals,'' *IEEE Trans. Acoust., Speech, Signal Processing* **ASSP-33 No.6** pp. 1404-1413 (December 1985).

[Juang86] B.H. Juang, S.E. Levinson, and M.M. Sondhi, ''Maximum likelihood estimations for multivariate mixture observations of Markov chains,'' *IEEE Trans, Informat. Theory* **IT-32, No.2** pp. 307-309 (March 1986).

[Juang87] B.H. Juang, L.R. Rabiner, and J.G. Wilpon, ''On the use of bandpass liftering in speech recognition,'' *IEEE Trans. Acoust., Speech., Signal Proccesing* **ASSP-35 No.7** pp. 947-954 (July 1987).

[Kampen81] N.G. Kampen, *Stocastic processes in Physics and Chemistry*, North-Holland, Elsevier Publishers (1981).

[Kohonen84] T. Kohonen, K. Makisara, and T. Saramaki, "Phonotopic maps, Insightful representation of phonological features for speech recognition," *IEEE 7 Conf. on Patt. Recognition, Proc.*, (1984).

[Kohonen84b] T. Kohonen, *Self-Organization and Associative Memory*, Springer-Verlag (1984).

[Kopec85] G.E. Kopec and M.A. Bush, "Network-based isolated digit recognition using vector-quantization," *IEEE Trans. Acoust., Speech, Signal Processing* **ASSP-33 No.4** pp. 850-867 (August 1985).

[Lamel81] L.F. Lamel, L.R. Rabiner, A.E. Rossenberg, and J.G. Wilpon, "An improved endpoint detector for isolated word recognition," *IEEE Trans. Acoust. Speech, Signal Processing* **ASSP-29 No.4** pp. 777-785 (August 1981).

[Lee88] C.H. Lee, F.K. Soong, and B.H. Juang, "A Segment Model Based Approach to Speech Recognition," *ICASSP'88*, pp. 501-504 (New York 1988).

[Lee86] K.F. Lee, "Incremental Network Generation in Word Recognition," *ICASSP'86*, pp. 77-80 (Tokyo 1986).

[Lee87] K.F. Lee, "Towards speaker-independent continuous speech recognition," *1987 NATO ASI on Speech Recognition an Dialog Understanding*, (1987).

[Lee88b] K.F. Lee, "Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System," *PhD Thesis*, Computer Science Department, Carnegie Mellon University, (1988).

[Lee89] K.F. Lee and H. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Trans. Acoust., Speech, Signal*

Processing **ASSP-37 No.11** pp. 1641-1648 (Noviembre 1989).

[Levinson83] S.E. Levinson, L.R. Rabiner, and M.M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *The Bell Syst. Tech. J.* **Vol.62 No.4** pp. 1025-1074 (April 1983).

[Linde80] Y. Linde, A. Buzo, and R.M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. Comm.* **COM-28, No.1** pp. 84-95 (January 1980).

[Liporace82] L.A. Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources," *IEEE Trans. Informat. Theory* **IT-28, No.5** pp. 729-734 (1982).

[Lippman87] R.P. Lippman, "An Introduction to Computing with Neural Nets," *IEEE Trans. Acoust. Speech, Signal Proc.* **ASSP-4, No.2** pp. 4-22 (April 1987).

[Lowerre80] B. Lowerre and R. Reddy, "The HARPY speech understanding system," *Trends in Speech Recognition*, pp. 340-346 Prentice-Hall, (1980).

[Lu91] N.A. Lu and D.R. Morrell, "VQ Codebook Design Using Improved Simulated Annealing Algorithms," *ICASSP'91*, pp. 673-676 (Toronto 1991).

[Makhoul85] J. Makhoul, S. Roucos, and H. Gish, "Vector Quantization in Speech Coding," *Procceeding of the IEEE* **Vol.73, No.11** pp. 1551-1588 (November 1985).

[Mariño90] J.B. Mariño, A. Bonafonte, A. Moreno, E. Lleida, C. Nadeu, and E. Monte, "Recognition of numbers by using demisyllabes and hidden

Markov models," *Proc. EUSIPCO'90* **Vol.2** pp. 1363-1366 (1990).

[Mariani87] J. Mariani, "Hamlet: A prototype of a Voice-Activated Typewriter," *European Conference on Speech Technology, Edinburgh*, pp. 222-225 (September 1987).

[Markel76] J.D. Markel and A.H. Gray, Jr., "The Linear Prediction Model," *Linear Prediction of Speech Comunications and Cybernetics* 12 pp. 1-18 Spriger Verlag. Berlin, Heidelberg, New York, (1976).

[Markel76b] J.D. Markel and A.H. Gray, Jr., "Spectral Analysis," *Linear Prediction of Speech. Comunications and Cybernetics* 12 pp. 129-163 Spriger Verlag. Berlin, Heidelberg, New York, (1976).

[Minsky69] M. Minsky and S. Papert, *Perceptrons: An introduction to Computational Geometry*, MIT Press (1969).

[Myers81] C.S. Myers and L.R. Rabiner, "A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition," *IEEE Trans. Acoust. Speech, Signal Proc.* **ASSP-29 No.2** pp. 284-297 (April 1981).

[Nadas81] A. Nadas, R.L. Mercer, L.R. Bahl, R. Bakis, P.S. Cohen, A.G. Coole, F. Jelinek, and B.L. Lewis, "Continuous Speech Recognition with Automatically Selected Acoustic Prototypes Obtained by Either Bootstraping or Clustering," *IEEE Int. Conf. on Acoustic, Speech, and Signal Processing*, (April 1981).

[Oppenheim75b] A.V. Oppenheim and R.W. Schafer, "Digital Filter Design Techniques," *Digital Signal Processing*, pp. 195-210 Prentice-Hall Inc., Englewood Clifss, New Jersey, (1975).

[Oppenheim75c] A.V. Oppenheim and R.W. Schafer, "The Z-Transform," *Digital Signal Processing*, pp. 45-86 Prentice-Hall Inc., Englewood Clifss, New Jersey, (1975).

[Oppenheim75] A.V. Oppenheim and R.W. Schafer, "Homomorphic signal processing," *Digital Signal Processing*, pp. 501-511 Prentice-Hall Inc., Englewood Clifss, New Jersey, (1975).

[Paliwal82] K.K. Paliwal, "On the performance of the quefrency-weighted cepstral coefficients in vowel recognition," *Speech Comm.* **Vol.1** pp. 151-154 (May 1982).

[Pan85] K.C. Pan, F.K. Soong, and L.R. Rabiner, "A vector-quantization-based preprocessor for speaker-independent isolated word recognition," *IEEE Trans. Acoust., Speech, Signal Proceesing* **ASSP-33 No.3** pp. 546-560 (June 1985).

[Peeling88] S.M. Peeling and R.K. Moore, "Isolated Digit Recognition Experiments Using the Multilayer Perceptron," *Speech Communications* **7, No.4** pp. 403-410 (December 1988).

[Peinado89] A.M. Peinado, "Reconocimiento Monolocutor de Palabras Aisladas Usando Modelos Ocultos de Markov y Cuantización Vectorial para Control de un Robot Mediante Voz," *Memoria de Licenciatura*, Dep. Electrónica y Tecnología de Computadores, Univ. de Granada, (1989).

[Peinado91] A.M. Peinado, J.M. López, V.E. Sánchez, J.C. Segura, and A.J. Rubio, "Improvements in HMM-based isolated word recognition system," *IEE Proc. I* **Vol.138, No.3** pp. 201-206 (June 1991).

[Peinado91b] A.M. Peinado, R. Román, J.C. Segura, A. Rubio, P. García, and J. Díaz, "Entropic training for HMM speech recognition," *Proc. EUROSPEECH-91*, (September 1991).

[Poritz82] A.B. Poritz, "Linear predictive hidden Markov models and the speech signal," *Proc. ICASSP'82 (Paris, France)*, pp. 1291-1294 (May 1982).

[Rabiner75] L.R. Rabiner and M.R. Sambur, "An algorithm for detecting the endpoints of isolated utterances," *Bell Syst. Tech. J.* **54 No.2** pp. 297-315 (February 1975).

[Rabiner78] L.R. Rabiner and R. Schafer, "Linear Predictive Coding of Speech," *Digital Signal Processing of Speech Signal*, pp. 396-461 Prentice-Hall, Inc. Englewood Cliffs, New Jersey, (1978).

[Rabiner78b] L.R. Rabiner and R. Schafer, "Short-Time Fourier Analysis," *Digital Signal Processing of Speech Signal*, pp. 250-354 Prentice-Hall, Inc. Englewood Cliffs, New Jersey, (1978).

[Rabiner83] L.R. Rabiner, S.E. Levinson, and M.M. Sondhi, "On the application of vector quantization and hidden Markov models to speaker-independent, isolated word recognition," *The Bell Syst. Tech. J.* **Vol.62 No.4** pp. 1975-1105 (April 1983).

[Rabiner84] L.R. Rabiner, "On the application of energy contours to the recognition of connected word sequences," *AT&T Bell. Lab. Tech. J.* **Vol.63 No.9** pp. 1981-1995 (November 1984).

[Rabiner85] L.R. Rabiner, B.-H. Juang, S.E. Levinson, and M.M Sondhi, "Recognition of isolated digits using hidden Markov models with continuous mixture densities," *AT&T Tech. J.* **Vol.64 No.6** pp. 1211-1233 (August 1985).

[Rabiner85b] L.R. Rabiner and F.K. Soong, "Single-Frame Vowel recognition using Vector Quantization with several distance measures," *AT&T Tech. J.* **Vol.64 No.10** pp. 2319-2331 (December 1985).

[Rabiner89] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications," *Proc. IEEE* **77 No.2** pp. 257-286 (February 1989).

[Rose90] K. Rose, E. Gurewitz, and G. Fox, "A Deterministic Annealing Approach to Clustering," *Pattern Recognition Letters* **11** pp. 589-594 Elsevier Science Publishers B.V. (North-Holland), (1990).

[Rosenblatt59] R. Rosenblatt, *Principles of Neurodynamics*, Spartan Books (New York 1959).

[Roucos82] S. Roucos, R. Schwartz, and J. Makhoul, "Segment Quantization for Very Low Rate Speech Coding," *ICASSP'82*, pp. 1565-1568 (Paris 1982).

[Rumelhart86] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing: Exploration in the Microstructure of Cognition* **1**MIT Press, (1986).

[Ruske81] G. Ruske and T. Schotola, "The Efficiency of Demi-Syllabe segmentation in the Recognition of Spoken Words," *ICASSP'81*, pp. 971-974 (Florida 1981).

[Ruske82] G. Ruske, "Auditory perception and its application to computer analysis of speech," *Computer Analysis and perception* **Vol.II,** *Auditory Signals*.C.Y. Suen and R. De Mori, Eds. Boca Raton, FL: CRC Press, (1982).

[Schwartz80] R.M. Schwartz, J. Klovstadt, J. Makhoul, and J. Sorensen, "A Preliminary Study of a Phonetic Vocoder Based on a Diphone Model," *ICASSP'80*, pp. 32-35 (Denver 1980).

[Segura84] J.C. Segura, "Cuatro Métodos de Codificación para el Reconocimiento de Palabras aisladas," *Memoria de Licenciatura*, Dept. Electrónica y Tecnología de Computadores. Univ. de Granada, (1984).

[Shakoe79] H. Shakoe, "Two-Level DP Matching - A Dynamic Programming Based Pattern Matching Algorithm for Connected Word Recognition," *IEEE Trans. Acoust. Speech, Signal Proc.* **ASSP-27 No.6** pp. 588-595 (December 1979).

[Shore82] J.E. Shore and D.K. Burton, "Discrete utterance speech recognition without time alignment," *ICASSP'82*, pp. 907-910 (May 1982).

[Shore83] J.E. Shore and D.K. Burton, "Discrete utterance speech recognition without time alignment," *IEEE Trans. Inform. Theory* **IT-29** pp. 473-491 (July 1983).

[Slutsker68] G.S. Slutsker, "Nelinejnyp Method Analiza Recevych Signalov," *Trudy Niir* **No.2**(1968).

[Soong88b] F.K. Soong and A.E. Rossenber, "On the use of instantaneous and transitional spectral information in speaker recognition," *IEEE Trans. Acoust., Speech, Signal Processing* **ASSP-36 No.6** pp. 871-879 (June 1988).

[Soong88] F.K. Soong and M.M Sondhi, "A frecuency weighted Itakura spectral distortion measure and its applications to speech recognition in noise," *IEEE Trans. Acoust., Speech, Signal Processing* **ASSP-36 No.1** pp. 41-48 (January 1988).

[Sugamura83] N. Sugamura, K. Shikano, and S. Furui, "Isolated Word Recognition Using Phoneme-Like Templates," *ICASSP'83*, pp. 723-726 (Boston 1983).

[Tohkura87] T. Tohkura, "A weighted cepstral distance measure for speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing* **ASSP-35 No.10** pp. 1414-1422 (October 1987).

[Torró-Enguix90] F. Torró-Enguix, E. Vidal, and H. Rulot, "Fast and accurate speaker-independent speech recognition using structural models learnt by the ECGI algorithm," *Proc. EUSIPCO'90* **Vol.2** pp. 1267-1270 (1990).

[Tou74] J.T. Tou and R.C. Gonzalez, "Pattern Classification by Distance Functions," *Pattern Recognition Principles*, pp. 75-109 Addison-Wesley Publishing Company, Inc., (1974).

[Vintsjuk68] T.K. Vintsjuk, "Recognition of Words of Oral Speech by Dynamic Programming," *Kibernetika* **81 No.8**(1968).

[Viterbi67] A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Trans. Informat. Theory* **IT-13** pp. 260-269 (April 1967).

[Waibel89] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme Recognition Using Time-Delay Neural Networks," *IEEE Trans. Acoust. Speech, Signal Processing*, (March 1989).

[Watanabe83] T. Watanabe, "Segmentation-free Syllabe Recognition in Continuous Spoken Japanese," *ICASSP'83*, pp. 320-323 (Boston 1983).

[Wilpon85] J.G. Wilpon and L.R. Rabiner, "A modified k-means clustering algorithm for use in isolated word recognition," *IEEE Trans. Acoust., Speech, Signal Processing* **ASSP-33 No.3** pp. 587-594 (June 1985).

[Young90] S.J. Young, "Competitive training in hidden Markov models," *Proc. ICASSP'90* **Vol.2** pp. 681-684 (1990).

DILIGENCIA:

Reunido el Tribunal examinador en el día de la fecha, constituido por:

D. Pedro Cartujo Estebanez

D. Climent Nadeu Camprubí

D. Francesc Casacuberta Nolla

D. Enrique Vidal Ruiz

D. Alberto Prieto Espinosa

para juzgar la Tesis Doctoral del Licenciado Don José Carlos Segura Luna

se acordó por unanimidad otorgar la calificación de Apto "cum laude"

y para que conste, se extiende firmada por los componentes del Tribunal, la presente diligencia.

Granada, a 22 de Noviembre de 1991

El Secretario,

[illegible] Alberto Prieto Espinosa

El Presidente,

Pedro Cartuj.

El Vocal,

[illegible]

El Vocal,

[illegible]

Francisco Casacuberta.

El Vocal,

[illegible]

Climent Nadeu