UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERIA ELÉCTRICA

ROBUSTEZ A VARIABILIDAD DE LOCUTOR EN RECONOCIMIENTO DE VOZ CON VTLN

IGNACIO CATALÁN LUDWIG

SANTIAGO DE CHILE
2011

UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERIA ELÉCTRICA

ROBUSTEZ A VARIABILIDAD DE LOCUTOR EN RECONOCIMIENTO DE VOZ CON VTLN

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELECTRICISTA

IGNACIO CATALÁN LUDWIG

PROFESOR GUÍA:
NÉSTOR BECERRA YOMA

MIEMBROS DE LA COMISION
CLAUDIO GARRETÓN VENDER
CARLOS MOLINA SÁNCHEZ

SANTIAGO DE CHILE
JUNIO 2011

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE:
INGENIERO CIVIL ELECTRICISTA
POR: IGNACIO CATALÁN L.
FECHA: 16/08/2011
PROF. GUÍA: SR. NÉSTOR BECERRA YOMA

"ROBUSTEZ A VARIABILIDAD DE LOCUTOR EN RECONOCIMIENTO DE VOZ CON VTLN"

El reconocimiento de voz (ASR, *Automatic Speech Recognition*) consiste en traducir a texto una señal de voz. Uno de los mayores problemas de los sistemas ASR son las variaciones en el locutor. La variabilidad entre las señales generadas por distintos hablantes al pronunciar una misma palabra es mucho mayor que la variabilidad entre señales de un único usuario pronunciando la misma palabra. Esto explica que los sistemas de ASR entrenados para un solo locutor tengan una tasa de aciertos superior a un sistema independiente del hablante.

El objetivo principal de la memoria es mejorar la robustez a la variabilidad de locutor en ASR. Para enfrentar este problema, una técnica ampliamente usada en la literatura es la normalización del largo del tracto vocal (VTLN, *Vocal Tract Lenght Normalization*). VTLN consiste en un ajuste (*warping*) del eje de frecuencias usado para parametrizar la señal de voz. Las funciones más usadas para realizar este ajuste dependen de un único parámetro. En aplicaciones típicas de VTLN es necesario hacer una búsqueda en barrido para poder encontrar el parámetro de normalización óptimo. En consecuencia se deben calcular las características de la señal para cada nivel de *warping* a evaluar, generando una carga computacional importante en los sistemas de ASR.

En esta memoria se propone una nueva técnica que modela el *warping* que se hace sobre el banco de filtros con VTLN como una interpolación lineal de energías de filtros vecinos. Este método, denominado IFE-VTLN, es comparado con un esquema estándar de VTLN. Con el procedimiento mencionado es posible encontrar el parámetro de normalización óptimo tanto mediante un barrido como analíticamente. Al usar el modo analítico, se mejora en más de 10 veces el tiempo requerido en comparación con VTLN estándar con optimización en barrido. Al usar la técnica propuesta con una búsqueda exhaustiva se obtienen disminuciones en el WER (*Word Error Rate*) de un 46.3% y un 38.7% cuando se compara con el sistema base y VTLN estándar, respectivamente. Al buscar analíticamente la solución se obtienen disminuciones en el WER (*Word Error Rate*) de un 31.3% cuando se compara VTLN estándar. Además se propone una extensión del esquema IFE-VTLN, llamado IFE-SA, en el cual se le añaden grados de libertad al modelo, permitiendo que cada filtro se interpole mediante un parámetro de ajuste. Es así necesario encontrar un vector de características de normalización óptimo, que solamente puede ser encontrado mediante un procedimiento analítico. Los resultados preliminares con IFE-SA muestran disminuciones en el WER de un 18.1% y 6.1% cuando se compara con el sistema base y VTLN estándar respectivamente. Se concluye que las técnicas propuestas son más eficientes que VTLN estándar tanto en reducción de WER como en eficiencia computacional.

Índice

1 Introducción

El reconocimiento de voz (ASR, *Automatic Speech Recognition*) consiste en traducir a texto una señal de voz. El número de aplicaciones que utilizan sistemas de reconocimiento automático de voz ha ido aumentando gradualmente con el tiempo, en la medida que ha ido mejorando el desempeño de estos sistemas por avances en investigación y en tecnología. Aún así, el problema de ASR está lejos de estar resuelto, y una de las grandes limitaciones que presentan es la a veces excesiva inferioridad del rendimiento de sistemas multi-locutor frente a sistemas mono-locutor. Por este motivo, el problema de la variabilidad de locutor ha sido ampliamente abordado en la literatura especializada, existiendo diversas técnicas y enfoques para disminuir este efecto.

1.1 Definición del Problema a Abordar

Para poder comprender el problema que es abordado en esta memoria, es necesario tener una noción básica de cómo se obtienen los parámetros acústicos de una señal. El primer paso de la parametrización acústica consiste en eliminar silencios y separar la señal acústica en *frames* o ventanas de tiempo (normalmente entre 20 y 30 milisegundos), etapa que será referida como pre-procesamiento. Las siguientes etapas son aplicadas a cada *frame* por separado. Primero se obtiene el espectro en frecuencia del *frame* aplicando la transformada de Fourier (FFT, *Fast Fourier Transform*). Luego se usa un banco de filtros, donde se calcula y extrae como parámetro la energía de la señal después de aplicar cada filtro. El vector de parámetros a la salida de esta etapa es un arreglo de las energías asociadas a cada filtro. En el siguiente bloque se aplica la función logaritmo sobre el vector de parámetros. Finalmente se aplica la transformada coseno discreta

(DCT, *Discrete Cosine Transform*) sobre el vector de parámetros y se obtiene lo que es llamado el vector de observación o parámetros cepstrales (MFCC, *Mel Frequency Cepstral Coefficients*). El detalle el proceso de parametrización es abordado en el siguiente capítulo de esta memoria. En la **Figura 1.1** se ilustra el proceso de obtención de parámetros acústicos mediante un diagrama de bloques.

Figura 1.1. Diagrama de bloques que describe el proceso de parametrización cepstral del *frame* de una señal de voz.

Una de las técnicas más conocidas y usadas en la literatura para disminuir el efecto de la variabilidad de locutor es la Normalización del Largo del Tracto Vocal (VTLN, *Vocal Tract Lenght Normalization*). VTLN intenta reducir los desajustes entre condiciones de entrenamiento y test en ASR, causados por la variabilidad intra-locutor como resultado de las diferencias en el largo del tracto vocal humano.

VTLN está definida como una modificación del eje del espectro de frecuencias en base a una función. Esta función depende de un factor que es llamado factor de *warping*. En el caso de usar un banco de filtros para parametrizar la señal –como se hace en esta memoria–, una forma de aplicar VTLN es usar la función que modifica el eje en frecuencias sobre las frecuencias centrales de los filtros del banco (Lee & Rose, 1998). Esta forma de normalización es la que será referida como VTLN convencional y es la técnica que suele usarse como base de comparación con respecto a avances que se publican en la literatura especializada.

A partir de la publicación de Pitz & Ney el 2005 (Pitz & Ney, 2005), donde se demostró que VTLN es equivalente a aplicar una transformada lineal (TL, transformada lineal) sobre el vector de observación, los últimos avances en VTLN se han centrado en transformaciones lineales del vector de observación (Cui & Alwan, 2006; Giuliani et. al., 2006; Panchapagesan & Alwan, 2009; Sanand et al., 2010; Umesh et. al., 2005; Wang et. al., 2007). La razón de intentar aplicar VTLN como una transformada lineal de los MFCC, es debido principalmente a la superioridad en velocidad de cómputo, al actuar sobre vectores de una dimensionalidad mucho más reducida que

en etapas anteriores. Además no es necesario realizar cómputos por los bloques de logaritmo y DCT que se deben realizar cuando se usa VTLN convencional.

Tanto para VTLN convencional como para las transformaciones lineales no es posible encontrar la normalización de parámetros óptima analíticamente, por lo que normalmente se hace una búsqueda en barrido para encontrar el factor de *warping* o la transformación lineal óptima. Esta forma de búsqueda es lenta, y para aplicaciones en tiempo real es deseable que la normalización de parámetros acústicos se realice en el menor tiempo posible.

En esta memoria se proponen dos nuevos algoritmos de normalización de parámetros acústicos. El primer método (IFE-VTLN, *Interpolation of Filter-bank Energies – VTLN*) se basa en una modelación de la modificación que realiza con VTLN sobre el banco de filtros. Como se verá más adelante, este método presenta ventajas como la sencillez de su aplicación, así como la posibilidad de obtener el factor de *warping* analíticamente, pudiéndose así disminuir drásticamente los tiempos de cómputo de la etapa de normalización. La segunda técnica (IFE-SA, *Interpolation of Filter-bank Energies – Spectral Alignment*) es una extensión a múltiples variables de IFE-VTLN. La forma de normalización de parámetros se aleja de la definición de VTLN, buscando más bien alinear el espectro de frecuencias de la señal de voz con el espectro del locutor de referencia (modelo acústico).

En la **Figura 1.2** se puede ver un diagrama de bloques de las etapas donde se hacen las modificaciones para implementar VTLN convencional (Lee & Rose, 1998), VTLN como transformación lineal del vector de observación (Cui & Alwan, 2006; Ding et. al., 2002; Giuliani et. al., 2006; Panchapagesan & Alwan, 2009, Pitz & Ney, 2005; Sanand et al., 2010; Umesh et. al., 2005; Wang et. al., 2007) y las técnicas IFE-VTLN e IFE-SA.

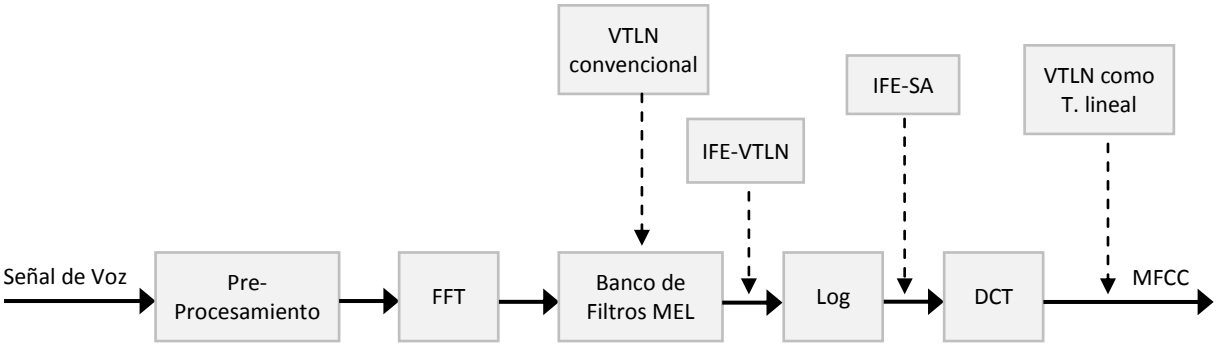


Figura 1.2. Diagrama de bloques que describe el dominio en el que actúan VTLN convencional, VTLN como transformación lineal, IFE-VTLN e IFE-SA. VTLN convencional se aplica cambiando el banco de filtros Mel, mientras que las otras técnicas son transformaciones lineales de los parámetros de salida de etapas distintas del proceso de parametrización.

1.2 Objetivos Generales y Específicos

1.2.1 Objetivo general

Mejorar la robustez a la variabilidad de locutor de los sistemas de reconocimiento de voz.

1.2.2 Objetivos específicos

- Proponer un método de normalización de parámetros acústicos para mejorar la robustez a la variabilidad de locutor de los sistemas de reconocimiento de voz.
- A diferencia de las técnicas convencionales de VTLN, la estimación de parámetros acústicos del método propuesto debe consistir en una optimización analítica.
- La técnica propuesta debe ser computacionalmente eficiente comparada con técnicas convencionales de VTLN.

2 Revisión Bibliográfica

2.1 Reconocimiento de Voz

En las últimas décadas se han experimentados grandes y rápidos cambios en el ámbito de la tecnología. La velocidad y capacidad de los sistemas informáticos aumenta exponencialmente, mientras los precios para acceder a diversas tecnologías son cada vez más bajos, produciéndose una masificación tecnológica en el mundo. Las interfaces que nos permiten interactuar con cualquier aparato tecnológico también han ido expandiéndose, siendo la voz una de estas interfaces que despierta gran interés, al tener ventajas como por ejemplo un número ilimitado (desde un punto de vista práctico) de instrucciones posibles al mismo tiempo de permitirnos tener nuestras manos ocupadas en alguna otra acción. Además es la voz la forma de comunicación principal entre seres humanos reunidos, pareciendo casi natural una extensión de esta forma de comunicación a la tecnología.

Las tecnologías de voz han ido aumentando su eficacia y confiabilidad a lo largo de los años, produciéndose un aumento en su uso para ciertas aplicaciones. Sin embargo aún existe una amplia gama de aplicaciones donde las tecnologías de voz no pueden ser aplicas por su falta de confiabilidad.

El reconocimiento de voz (ASR) es una de las tecnologías más desarrolladas dentro del área de la voz y consiste en traducir a texto a una señal de voz. Desde el punto de parámetros acústicos los sistemas de reconocimiento de voz se ven fuertemente afectados por el ruido ambiental, el ruido de canal y la variabilidad de locutor.

2.1.1 Formulación del problema

El reconocimiento automático de voz es un proceso de reconocimiento de patrones. Esto implica que se debe extraer la información de las señales acústicas en forma de un conjunto de clases conocidas. Para resolver el problema de ASR con un enfoque estadístico se emplea el teorema de Bayes, modelando el proceso de reconocimiento de voz como un problema de maximización a posteriori. La probabilidad de que se diga una secuencia de palabras W y se extraiga de la señal acústica la secuencia de vectores de parámetros O, puede reescribirse usando Bayes como:

$$P(W,O) = P(W/O)\cdot P(O) \tag{2.1}$$

donde $W = \{w_1, w_2, ..., w_J\}$ representa a la secuencia de palabras y $O = \{o_1, o_2, ..., o_T\}$ corresponde a la secuencia de vectores de parámetros generados a partir de la señal acústica.
Al aplicar Bayes nuevamente se obtiene:

$$\begin{aligned} P(W/O)\cdot P(O) &= \left(\frac{P(O/W)\cdot P(W)}{P(O)}\right)\cdot P(O) \\ &= P(O/W)\cdot P(W) \end{aligned} \tag{2.2}$$

Se desea encontrar la mejor secuencia de palabras W, dado una secuencia de vectores de parámetros O. Esto se traduce en maximizar la probabilidad $P(O/W)$. Es así como finalmente la decisión en los ASR será para aquel conjunto de palabras que maximice la siguiente ecuación (Jelinek, 1997) (Rabiner et. al., 1996):

$$\hat{w} = \arg\max_{w} P(W/O) = \arg\max_{w} P(O/W)\cdot P(W) \tag{2.3}$$

La primera expresión del argumento de la maximización de la ecuación (2.3), es decir $P(O/W)$, entrega la probabilidad de que dada una secuencia de palabras, denotadas por W, haya generado una secuencia de vectores de parámetros O. Esta probabilidad se conoce como el modelo acústico del reconocedor de voz. El segundo término representa a la ocurrencia de las clases (palabras), lo que es conocido como el modelo de lenguaje del sistema.

2.1.2 Medida de desempeño del reconocedor

Una métrica usada comúnmente en la literatura para evaluar el rendimiento de un reconocer es la tasa de palabras erradas o mal clasificadas o WER (*Word Error Rate*) definido como:

$$WER = \frac{S + I + D}{N} \cdot 100 \tag{2.4}$$

donde N corresponde al número total de palabras de test, es decir las palabras que efectivamente fueron pronunciadas, y S, I y D son el número de palabras sustituidas, insertadas y eliminadas, respectivamente.

Además se medirá la reducción porcentual del WER con las técnicas de adaptación con respecto al WER del *baseline* (caso base del experimento, sin aplicar las técnicas de adaptación). Esto se expresa como:

$$\text{Reducción Porcentual del } WER = \frac{WER_{baseline} - WER_{\exp}}{WER_{baseline}} \cdot 100 \tag{2.5}$$

donde WERexp corresponde al WER del experimento de reconocimiento usando alguna técnica de adaptación.

2.2 Técnicas Usadas en ASR

2.2.1 Parametrización acústica

El método usado para la parametrización de señales de voz se basa en el cálculo de coeficientes *cepstrales*. Analizar una señal de voz en el dominio cepstral o *cepstrum* contribuye a realzar las componentes asociadas a los formantes del tracto vocal, incluso en señales con ruido. En la **Figura 1.1** se puede apreciar el proceso de extracción de características acústicas.

Los parámetros basados en el *cepstrum* se han convertido en uno de los métodos más usados en clasificación de patrones acústicos y ya se ha transformado en un estándar dentro del área de procesamiento de voz (Forsyth, 1995). Generalmente a las señales de voz se les hace un pre-procesamiento con el objeto de realzar la información de voz por sobre otro tipo de información que pueda contener la señal. Esto permite que las señales se encuentren en condiciones similares para la obtención de parámetros. Lograr esta homogenización de las señales se puede lograr mediante las siguientes tareas: detección del inicio y fin de la información de voz; supresión de segmentos de silencio; y, compensación de ruido aditivo y/o convolucional.

En la primera etapa del pre-procesamiento la señal es tratada por un filtro inicio-fin el que elimina la información irrelevante que esta antes y después del primer y último pulso de voz detectados (Lamel et al., 1981; Savoji, 1989). Luego se divide la señal en segmentos que se denominan ventanas o *frames*. Los *frames* son la unidad básica para la posterior caracterización de la señal (obtención de parámetros). Para esta segmentación generalmente se toman intervalos de 10 a 30 [mseg] y con traslapes entre ventanas consecutivas. El traslape entre ventanas consecutivas puede llegar a ser hasta un 60%. Por último se utiliza la técnica de enventanado de *Hamming* (Picone, 1993), para evitar las distorsiones en el análisis espectral que son generadas por ventanas rectangulares.

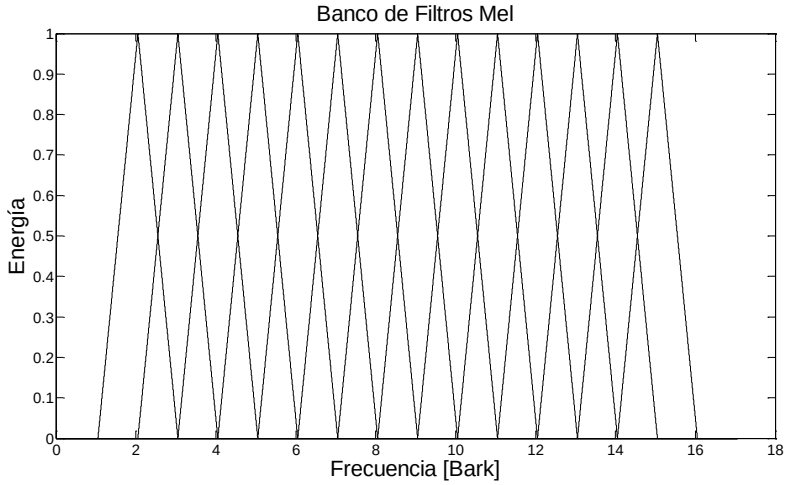
La etapa de parametrización se realiza sobre cada ventana por separado. Al final de esta etapa se obtendrá por lo tanto un vector de parámetros por cada *frame*. El proceso comienza con un análisis espectral del *frame*, para lo cual es aplicada la transformada rápida de Fourier (FFT, *Fast Fourier Transform*). Una vez que se tiene el espectro de frecuencias, se aplica un banco de filtros que consta de un filtro por cada banda de interés. La aplicación del banco de filtros consiste en:

filtrado del espectro con cada filtro; cálculo de la energía del espectro resultante del filtrado. Se tendrá entonces una energía asociada a cada filtro del banco. Solamente se usarán estas energías para lo que sigue en el proceso de obtención de parámetros. (Las energías obtenidas de la aplicación del banco de filtros serán referidas como "energías de los filtros" o "energía del filtro" si se refiera a 1 filtro individual, lo cual es semánticamente erróneo pero simplifica la redacción). El banco de filtros se utiliza dado que la percepción auditiva humana no es capaz de distinguir frecuencias individuales, sino que capta franjas de frecuencias. Las bandas que captura el banco de filtro están entre los 300 y 3400 [Hz], que es el rango de frecuencias de interés, dado que contiene la información más relevante que permite reconocer el habla. Además la respuesta del sistema auditivo humano en el espectro de frecuencias no es lineal, para lo cual se han creado escalas que representan este comportamiento. Una de las escalas más utilizadas para estos efectos son las escalas Mel y Bark. En (2.6) y (2.7) se describen las transformaciones asociadas a las escalas Mel y Bark respectivamente, para un valor de frecuencia f :

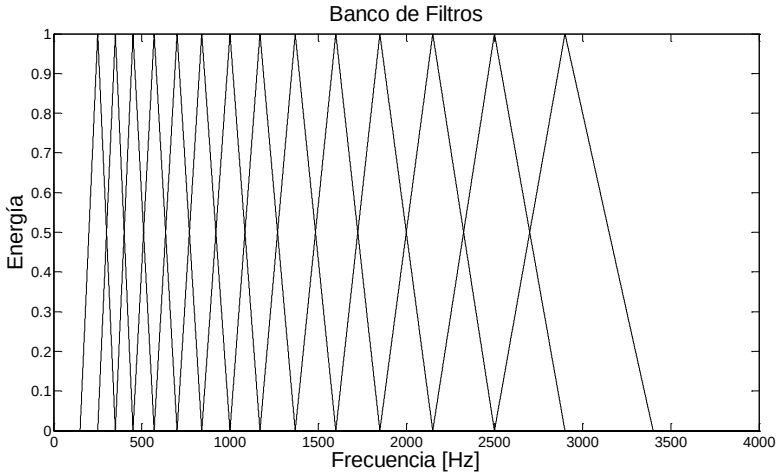
$$Mel(f) = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right); \quad f \text{ en Hertz} \tag{2.6}$$

$$Bark(f) = 13 \cdot \arctan(0.00075 \cdot f) + 3.5 \cdot \arctan((f/7500)^2) \tag{2.7}$$

Un ejemplo de banco de filtro se puede apreciar en la **Figura 2.1** (a), que se como se observa está compuesto por una serie de filtros triangulares en escala Bark. Notar que se denominan banco de filtros Mel a pesar de estar en escala Bark, dado que este banco de filtros fue creado originalmente en escala Mel (ambas escalas son muy similares). Los filtros tienen todos la misma ganancia *peak*, son simétricos, tienen una superposición de 50% y un ancho de banda constante en escala Bark. En la **Figura 2.1** (b) se ilustra el mismo banco de filtros pero en escala lineal.



(a)



(b)

Figura 2.1. En (a) se puede apreciar un banco de filtros compuesto por 14 filtros triangulares de ancho de banda constante en escala Bark. En (b) se puede ver el mismo banco de filtros pero para frecuencias en Herz. Se observan filtros de ancho de banda creciente a medida que aumenta la frecuencia, lo que obedece a la respuesta no lineal del sistema auditivo humano.

El siguiente paso consiste en calcular el logaritmo de la energía de cada filtro, obteniéndose así las características MFBLE (*Mel-Filterbank Log-Energy*). El último paso consiste en realizar el cálculo de coeficientes cepstrales en escala Mel (MFCC, *Mel Frequency Cepstral Coefficient*). Para esto esto se usa la transformada coseno discreta (DCT, *Discrete Cosine Transform*) sobre los parámetros MFBLE. Como se explicó anteriormente, este proceso se debe realizar sobre cada *frame* a analizar, por lo que se obtiene un vector de parámetros MFCC para cada *frame*, es decir,

una señal de voz es caracterizada como una secuencia de vectores de observación en el dominio MFCC.

Como se puede ver en (Bimbot et al., 2004, Furui, 2005) el uso de las características basadas en MFCC es predominante en las áreas de reconocimiento de voz/locutor y se mantiene relativamente invariante.

2.2.2 Modelamiento acústico con modelos ocultos de Markov

Las cadenas de Markov consisten en una secuencia finita de estados interconectados por probabilidades de transición. Cada estado tiene una función de distribución de probabilidad la cual entrega la verosimilitud de que una observación haya sido generada por él (Rabiner, 1989). Los modelos ocultos de Markov (HMM, *Hidden Markov Models*), han sido ampliamente utilizados en los sistemas de reconocimiento de voz y locutor. En particular, los modelos más usados son los de de primer orden, donde el estado al que se asocia cada *frame* depende del vector de parámetros que lo caracteriza y del estado asociado al *frame* anterior (Rabiner, 1989; Jelinek, 1997).

Una secuencia de estados genera una secuencia de vectores de parámetros. Lo que se observa son los vectores de parámetros, mientras que la secuencia de estados que los generó es desconocida. Existen varias combinaciones de secuencias posibles que pueden generar la pronunciación de las mismas palabras y no se supone ninguna secuencia de estados única durante el proceso. Todas las secuencias son consideradas y por lo tanto la secuencia de estados real permanece oculta. Lo que sí se conoce (hay que entrenar un modelo) son las probabilidades de que los estados generen cualquier observación y las probabilidades de transición de estados, que es la información que se usará para calcular la probabilidad de que una cierta secuencia de estados genere las observaciones.

La topología usada en HMM aplicado a ASR se denomina "*left-to-right*", es decir, permiten sólo transiciones al siguiente o al mismo estado. Con esto se limitan los saltos o retrocesos. Un HMM queda definido por: las probabilidades de transición de estados, la función de distribución de

probabilidad y las probabilidades iniciales (Rabiner, 1989). Las probabilidades de transición para un HMM con M estados debe cumplir con la siguiente restricción:

$$\sum_{j=1}^{M} A(i,j) = 1 \qquad \forall i = 1,...,M \tag{2.8}$$

donde $A(i,j)$ corresponde a la probabilidad de estar en el estado j dado que el anterior estado fue i. La distribución de probabilidad de que una observación haya sido generada por el estado j se representa en (2.9). Cabe mencionar que en la tarea de reconocimiento de voz es usual utilizar poblaciones para modelar las f.d.p. de estados. En este caso suponemos una población de G distribuciones normales independientes, cada una con un peso de probabilidad asignado, y restringido por (2.10):

$$b_j(O_t) = \sum_{g=1}^{G} \left\{ p_g \cdot \prod_{n=1}^{N} \left[(2\cdot\pi)^{-0.5} \cdot \left(Var_{j,g,n}\right)^{-0.5} \cdot e^{-\frac{1}{2}\frac{\left(O_{t,n}^{o} - E_{j,g,n}\right)}{Var_{j,g,n}}} \right] \right\} \tag{2.9}$$

$$\sum_{g=1}^{G} P_g = 1 \tag{2.10}$$

donde j,g,n son los índices para el estado, la componente Gaussiana y el coeficiente del vector de observación, respectivamente; P_g corresponde al peso de probabilidad de la población g-ésima; $O_t = \left[O_{t,1}^{o}, O_{t,2}^{o}, ..., O_{t,N}^{o}\right]$ es el vector de observación de la señal acústica de dimensión N en el instante t; $E_{j,g,n}$ y $Var_{j,g,n}$ son la media y varianza para un determinado modelo en el estado j, componente Gaussiana g y coeficiente cepstral n. Cabe mencionar que la matriz de covarianza de las Gaussianas es supuesta diagonal, es por esta razón que se hace mención a la varianza.

Un HMM representa una unidad fonética. En este caso se utilizan los denominados trifonemas, estos se componen de una unidad fonética (formantes) central más dos segmentos de fonemas

que preceden y suceden a la unidad central (Schwartz et. al., 1985). Cualquier palabra o frase puede ser generada a partir de una secuencia de trifonemas. Así, es posible deducir la probabilidad de que la secuencias de vectores de parámetros acústicos O haya sido generada por el HMM de la secuencia de palabras W, siendo:

$$P(O/W) = \sum_{S \in \Lambda} P(O,S/W) = \sum_{S \in \Lambda} P(S/W) \cdot P(O/S) \tag{2.11}$$

donde $S = [s_1, s_2, ..., s_T]$ representa cualquier secuencia de estado dentro del conjunto Λ; el conjunto Λ son todas las posibles secuencias de estados que pueden generar W. En la práctica, calcular todas las posibles secuencias resulta muy costoso computacionalmente, por lo que suele usarse calcularse (2.11) usando solamente la secuencia de estados que maximiza las probabilidades de observación. Usando esta aproximación y descomponiendo según la descripción de un HMM y reemplazar en (2.3), se obtiene:

$$\begin{aligned} \hat{W} &= \arg\max_{w,S} \{P(W) \cdot P(O/W)\} \\ &= \arg\max_{w,S} \left\{ P(W) \cdot \left(A(0,S) \cdot \prod_t A(S_t, S_{t+1}) \right) \cdot \left(\prod_t b_{S_t}^W (O_t) \right) \right\} \end{aligned} \tag{2.12}$$

Un ejemplo de arquitectura HMM se muestra en la **Figura 2.2**.

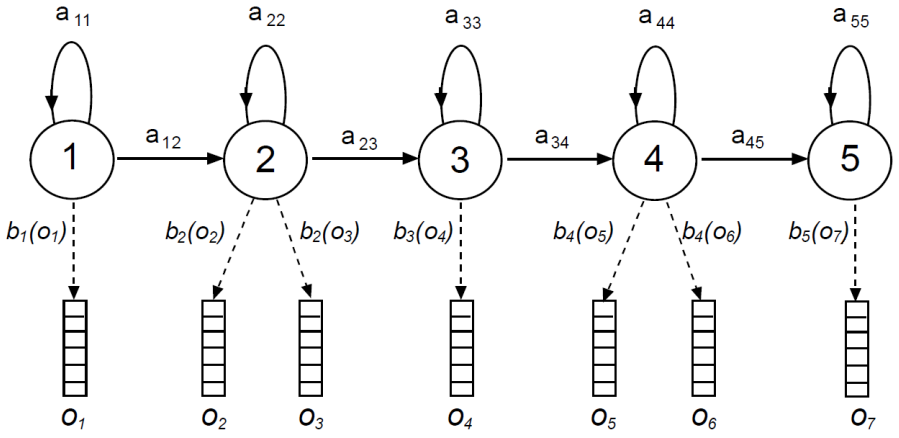


Figura 2.2. Ejemplo de topología izquierda derecha sin salto de estado de un HMM.

2.2.3 Modelo del lenguaje

El modelo de lenguaje entrega información a priori en la tarea de reconocimiento de la voz, $P(W)$ en (2.3). Los métodos para estimarlo pueden variar desde ser un algoritmo de reglas gramaticales, hasta ser netamente una representación estadística del lenguaje utilizado. Los más usados son los modelos estocásticos de tipo M-grama. Esto considera que la ocurrencia de una palabra dentro de una sucesión de ellas está condicionada a la probabilidad de las M-1 palabras anteriores. Un modelo M-grama se representa como:

$$P(w_1, w_2, w_3, ..., w_N) = \prod_{i=1}^{N} P(w_i | w_{i-M+1}, ..., w_{i-2}, w_{i-1}) \tag{2.13}$$

El criterio para la estimación de los parámetros que determinan el modelo de lenguaje es el estimador de máxima verosimilitud. En él se maximiza la probabilidad de observar las secuencias de algún conjunto de entrenamiento.

Uno de los problemas de los modelos estocásticos es que no considera probabilidad para las secuencias de palabras que no se encuentran en el conjunto de entrenamiento. Según la definición, estas probabilidades quedan en cero para aquellos casos en que no existe ocurrencia. El problema de generalización del modelo de lenguaje es tratado con diversas técnicas. Por ejemplo, existe el modelo de lenguaje a nivel de clases, depuración de parámetros o modelos de lenguaje por palabras (Laurila et. al., 1998; Becchetti & Prina, 1999).

Lo que aún falta por resolver es cómo encontrar la secuencia de estados óptima que genera un vector de parámetros acústicos. Para resolver este problema su usa el algoritmo de Viterbi.

2.2.4 Algoritmo de Viterbi

Para encontrar la secuencia de estados óptima en (2.12) se podrían evaluar todas las posibles secuencias de estado para cada instante de tiempo en la señal de voz. Sin embargo el número de secuencias posibles crece exponencialmente con el largo total de la secuencia, siendo

impracticable realizar una búsqueda de este tipo. El problema puede ser resuelto mediante el algoritmo de Viterbi. Este método consiste en ir optimizando a nivel local las secuencias de estado. Con ello, en forma inductiva, se resuelve el problema de optimización global (Jelinek, 1997). El algoritmo de Viterbi al optimizar a nivel local va descartando secuencias, logrando reducir el campo de búsqueda para que el problema sea viable desde el punto de vista computacional.

Sea $\delta_t(i)$ la probabilidad de observar la secuencia de parámetros O hasta el tiempo t junto con la secuencia de estados más verosímil hasta t y que además, el estado s en t sea i:

$$\delta_t(i) = \max_{s_1, s_2, ..., s_{t-1}} P(s_1, s_2, ...s_{t=1}, s_t = i, o_1, o_2, ..., o_t / \lambda_w) \tag{2.14}$$

Suponiendo recursividad se obtiene:

$$\begin{aligned} \delta_t(i) &= \max_{s_1, s_2, ..., s_{t-1}} P(o_t, s_t = i / s_1, ..., s_{t-1}, o_1, ..., o_{t-1}, \lambda_w) \cdot P(s_1, ..., s_{t-1}, o_1, ..., o_{t-1} / \lambda_w) \\ &= b_i(o_t) \cdot \max_{s_1, s_2, ..., s_{t-1}} P(s_1, s_2, ..., s_{t-1}, o_1, o_2, ..., o_{t-1} / \lambda_w) = b_i(o_t) \cdot \max_{s \in \Gamma} \left(a(s,i) \cdot \delta_{t-1}(s) \right) \end{aligned} \tag{2.15}$$

Para llegar al cálculo de $\delta_t(i)$ se debe evaluar todos los posibles caminos para llegar a $s_t = i$. Estos posibles caminos están agrupados en el espacio Γ, por lo tanto Γ es un conjunto de secuencias de t estados, es decir $\Gamma \in \Re^t$. λ_w es el modelo de la secuencia de palabra W hasta el instante t. El término $a(s,i)$ determina la probabilidad de transición del último estado en la secuencia S al estado dado en t que es i. Asumiendo la recursividad del algoritmo, si buscamos la secuencia de estados más verosímil para llegar a s_t, la secuencia anterior debe ser $\delta_{t-1}(s)$ donde s pertenece al conjunto Γ. Con esto, en forma recursiva, se llega a que $\delta_t(i)$ es la secuencia más probables de estados para llegar al tiempo t con el estado i. Luego, para obtener la información del estado en el cual se está en el tiempo t se define la función $\psi_t(i)$, que a medida que se avanza en el algoritmo guardará la información del estado óptimo. Finalmente, el algoritmo se define según la secuencia que se describe a continuación.

1. Inicialización:

$$\begin{aligned} &\delta_1(i) = \pi_i \cdot b_1(o_1) \qquad i \in \Gamma \\ &\psi_1(i) = 0 \end{aligned} \tag{2.16}$$

2. Recursión:

$$\begin{aligned} &\delta_t(i) = b_i(o_t) \cdot \max_{s \in \Gamma}\left(a(s,i) \cdot \delta_{t-1}(s)\right) \qquad i \in \Gamma 2 \le t \le k \\ &\psi_t(i) = \arg\max_{s \in \Gamma}\left(a(s,i) \cdot \delta_{t-1}(s)\right) \qquad i \in \Gamma 2 \le t \le k \end{aligned} \tag{2.17}$$

3. Finalización, se determina la probabilidad de la secuencia de estados más verosímiles y el último estado de dicha secuencia:

$$P_{\max} = \max_{i \in \Gamma}\left(\delta_k(i)\right) \quad ; \quad \hat{s} = \arg\max_{i \in \Gamma}\left(\delta_k(i)\right) \tag{2.18}$$

4. Alineamiento: se reconstruye la secuencia de estados más verosímil.

$$s_t = \psi_{t+1}(s_{t+1}) \qquad t = 1, \ldots, k-1 \tag{2.19}$$

2.3 Normalización del Largo del Tracto Vocal (VTLN)

La variabilidad temporal en las señales de voz puede deberse a factores relacionados con el locutor, el entorno y la fuente o medio de captura de la voz. El factor relacionado con el locutor puede separarse en dos; variabilidad intra-locutor, que se describe como las variación entre elocuciones de un mismo individuo de la información acústico fonética se que extrae de la señal voz; variabilidad inter-locutor, que se describe como las variaciones entre elocuciones pertenecientes a un grupo amplio (o universo) de locutores. Otro factor que puede introducir una

componente de variabilidad no deseada al momento de parametrizar una señal de voz, es la cantidad de ruido ambiental y la variabilidad de este en el tiempo. Finalmente se tiene el efecto del medio de captura de la voz o canal de comunicación, que puede generar fuertes distorsiones en elocuciones con idéntica información fonética de un mismo usuario.

Como es de esperarse, los factores de variabilidad mencionados causan degradación en los sistemas de ASR y toman más o menos relevancia dependiendo de la aplicación de estos sistemas.

Uno de los problemas determinantes en todo sistema de ASR que opera en entornos con múltiples locutores son las variaciones inter-locutor. La variabilidad entre las señales generadas por distintos locutores al pronunciar una misma palabra es mucho mayor que la variabilidad entre señales de único locutor pronunciando la misma palabra. Esto explica que los sistemas de reconocimiento de voz entrenados para un único locutor tengan una tasa de aciertos superior a un sistema independiente de locutor.

La idea de ajustar el eje de las frecuencias de señales de voz para hacer frente a las variaciones propias del género en el reconocimiento de vocales aisladas fue propuesta por primera vez en (Wakita, 1977). Esta idea fue recogida más adelante en (Acero, 1990; Acero and Stern, 1991) para pequeños vocabularios. Para grandes vocabularios, la Normalización del Largo del Tracto Vocal (VTLN, *Vocal Tract Length Normalization*) ha sido propuesta en (Eide & Gish, 1996; Lee & Rose, 1998; Wegmann et. al., 1996). En (Eide & Gish, 1996) son comparadas varias funciones de *warping* (ajuste), obteniéndose resultados similares en cuanto a porcentajes de mejoras en el reconocimiento. Una estimación basada en el criterio de máxima verosimilitud fue sugerida por (Lee & Rose, 1998), junto con un esquema iterativo para estimar el factor de *warping*. La estimación de los factores de *warping* es hecha por un alineamiento forzado usando un reconocimiento preliminar o por un modelo de mezcla de Gaussianas sin la necesidad de un reconocimiento preliminar. Se logran mejoras relativas en el WER de un 20% y 15% con y sin reconocimiento preliminar respectivamente. Un enfoque similar fue presentado en (Wegmann et.

al., 1996), donde se aplica una función de *warping* lineal por partes, lográndose mejoras relativas de un 12% en el WER.

Modelar la función de *warping* en el dominio espectral y como transformación lineal en el dominio cepstral, ha sido propuesto por algunos autores (Claes et. al., 1998; Cui & Alwan, 2006; Ding et. al., 2002; Giuliani et. al., 2006; Panchapagesan & Alwan, 2009, Pitz & Ney, 2005; Sanand et al., 2010; Umesh et. al., 2005; Wang et. al., 2007). Aquellas técnicas pueden representar la función de *warping* espectral en el dominio cepstral como: una transformación lineal en el dominio cepstral continuo (Claes et. al., 1998; Ding et. al., 2002; Pitz & Ney, 2005; Sanand et. al., 2010; Umesh et. al., 2005) ; y, un caso particular *de Maximum Likelihood Linear Regression* (MLLR) (Leggetter & Woodland, 1995) en el dominio cepstral discreto (Claes et. al., 1998; Cui & Alwan, 2006; Ding et. al., 2002; Giuliani et. al., 2006; Panchapagesan & Alwan, 2009, Sanand et al., 2010; Umesh et. al., 2005; Wang et. al., 2007). En el primer grupo de técnicas, la búsqueda por barrido de MV aún es el procedimiento más común usado para obtener el factor de *warping* óptimo. Al contrario, en el segundo grupo de métodos es posible obtener el factor de *warping* empleando un procedimiento de optimización analítico. Por ejemplo, en (Claes et. al., 1998; Cui & Alwan, 2006; Ding et. al., 2002; Giuliani et. al., 2006; Panchapagesan & Alwan, 2009, Sanand et al., 2010; Umesh et. al., 2005; Wang et. al., 2007), la optimización se realiza usando el algoritmo de *Expectation-Maximization (EM).*

2.3.1 Ajuste o warping del banco de filtros

La distorsión del eje del espectro de frecuencias se realiza considerando una función de *warping*. Una de las funciones más usadas corresponde a la función lineal por tramos (Pitz et al., 2001), que se muestra en la **Figura 2.3**.

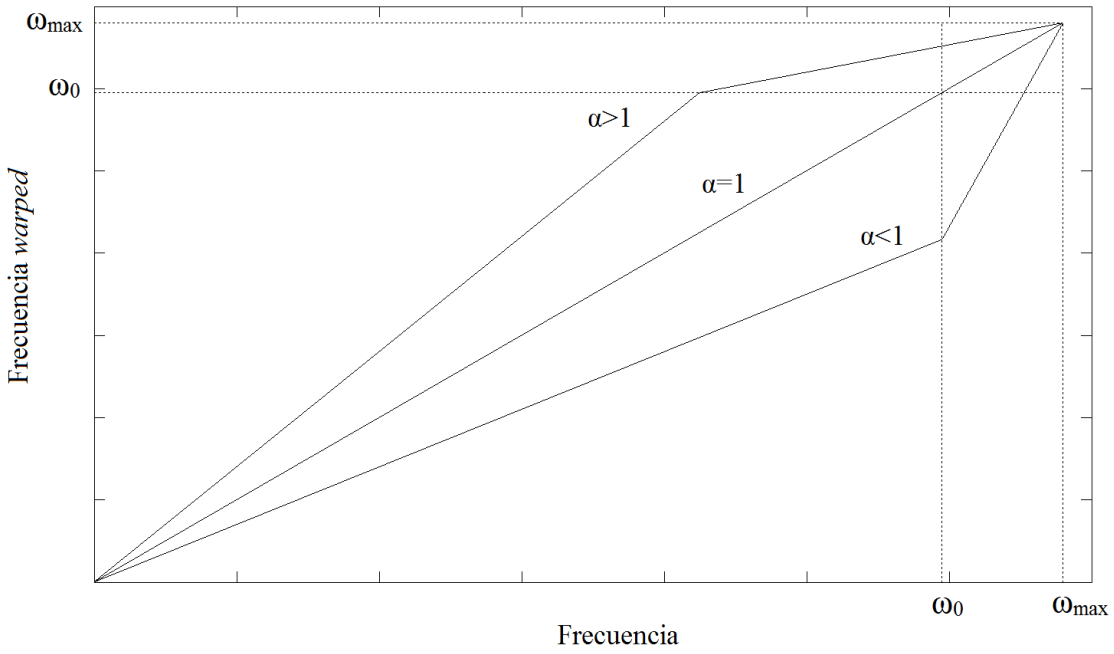


Figura 2.3. Función lineal por tramos para distintos valores de α, siendo $\omega_0 = \frac{7}{8}\omega_{\max}$. Notar que $\alpha < 1$ corresponde a comprimir el espectro de frecuencias, $\alpha = 1$ es el caso sin distorsión y $\alpha > 1$ corresponde a estirar el espectro. Dado que el largo del tracto vocal de las mujeres es menor que en los hombres, resulta necesario estirar el espectro y, por ende, típicamente a es mayor que uno. En hombres ocurre el efecto opuesto.

Esta función se define como:

$$g_{(\alpha,\omega_0)} : \omega \to \tilde{\omega} = \begin{cases} \alpha \cdot \omega & si\ \omega \le \omega_0 \\ \alpha \cdot \omega_0 + \frac{\omega_{\max} - \alpha\omega_0}{\omega_{\max} - \omega_0} \cdot (\omega - \omega_0) & si\ \omega > \omega_0 \end{cases} \tag{2.20}$$

donde ω_0 es el punto de inflexión donde cambia la pendiente de la función, el que se define como:

$$\omega_0 = \begin{cases} \frac{7}{8}\omega_{\max} & \alpha \le 1 \\ \frac{7}{8 \cdot \alpha}\omega_{\max} & \alpha > 1 \end{cases} \tag{2.21}$$

Cuando se usa un banco de filtros para parametrizar la señal, resulta equivalente generar la distorsión del espectro de frecuencias sobre el banco de filtros en vez de la señal (Lee & Rose, 1998). En esta memoria se hará *warping* solamente sobre las frecuencias centrales y de los extremos de las frecuencias del banco de filtros Mel (Lee & Rose, 1998; Panchapagesan & Alwan, 2009) como se ve en la **Figura 2.4**. Esta es una variante de VTLN, que permite que los filtros mantengan su forma triangular. Si se aplica *warping* sobre todo el espectro de frecuencias del banco de filtros, los filtros se deformaran (los lados de los triángulos toman curvatura).

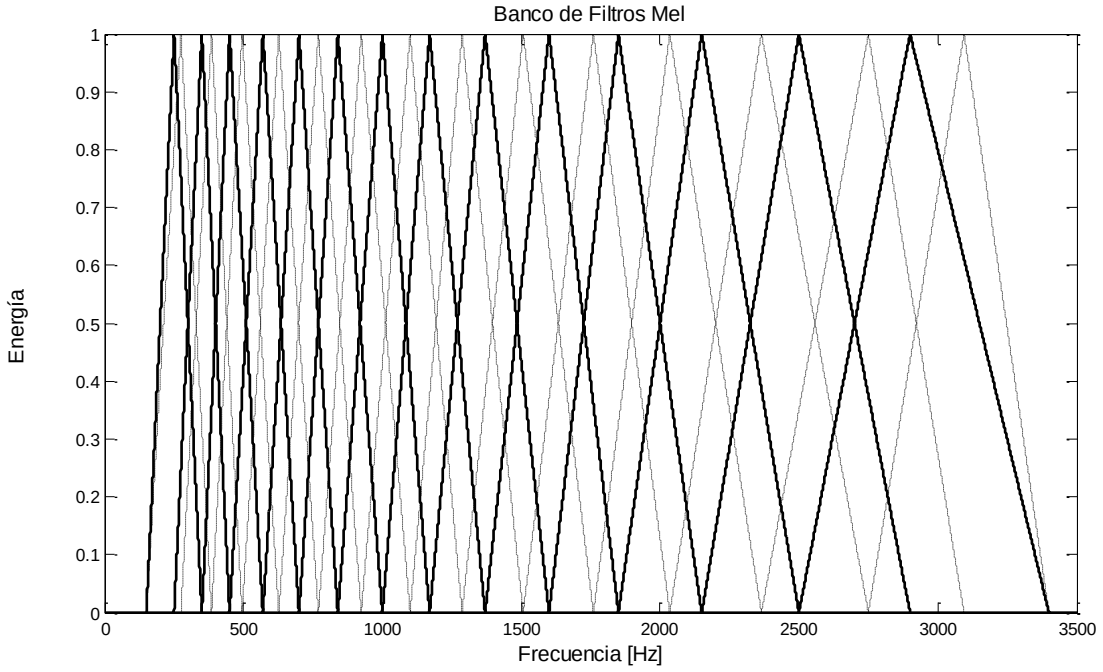


Figura 2.4. Banco de filtros Mel original (línea continua) y *warped* para un $\alpha > 1$ (línea segmentada).

2.3.2 Factor de ajuste óptimo

Sea $\hat{\alpha}_i$ el factor de normalización asociado a un locutor i. Sea I un conjunto de HMMs, X_i^{α} corresponde a un conjunto de coeficientes cepstrales de un conjunto de elocuciones normalizados por el factor, α y W denota las transcripciones de las elocuciones. Entonces el factor de normalización óptimo de un locutor i se define como:

$$\hat{\alpha}_i = \arg\max_{\alpha} P\left(X_i^{\alpha} / \lambda, W_i\right) \tag{2.22}$$

En general resulta difícil encontrar una solución exacta de la ecuación (2.22) por lo que típicamente se realiza un barrido, es decir una búsqueda exhaustiva considerando un conjunto finito de factores de normalización. Por ejemplo, en (Lee & Rose, 1998) se usan valores entre 0,88 y 1,12 (*line search*) donde 1 equivale al caso sin normalización.

Uno de los problemas de VTLN es que para realizar una búsqueda del factor mediante un barrido (*line search*), se necesitan calcular las nuevas características normalizadas de la señal para cada uno de los factores del barrido. Las técnicas más recientes se basan en el hecho de que el *warping* en frecuencias es equivalente a una transformada lineal en el espacio cepstral *(*Pitz et al., 2001; Panchapagesan & Alwan, 2009). Esto permite realizar el proceso de VTLN de forma menos costosa computacionalmente, dado que la transformada puede ser aplicada directamente sobre las características obtenidas (MFCC).

El método que se propone en esta memoria tiene una solución aproximada que se puede encontrar analíticamente, que lo diferencia los métodos típicos de VTLN que requieren de un barrido para encontrar el factor óptimo.

3 Normalización de locutor en el dominio Cepstral con optimización analítica aplicado a ASR

3.1 VTLN con un mediante interpolaciones del vector de observación en ASR

En esta sección se propone un método donde las energías del banco de filtros *warped* se estiman haciendo uso de una interpolación lineal entre energías de filtro contiguas en los bancos de filtros originales. Por otra parte, un esquema de optimización analítica para obtener el factor de *warping* óptimo se deriva para reemplazar la búsqueda en barrido. El aporte de esta sección de la memoria se ocupa de: a) un modelo VTLN en el dominio energético del banco de filtros basado en la interpolación de energías de banco de filtro (IFE-VTLN, *Interpolation of Filter Energies* - VTLN); y, b) una estimación analítica basada en el criterio de máxima verosimilitud (MV) del factor de *warping* óptimo acorde al modelo IFE-VTLN.

3.1.1 Interpolación del vector de observación

Considere que ω_m es la frecuencia central del filtro m en un banco de filtro compuesto por M filtros. Luego $\hat{\omega}_m$ es la frecuencia central *warped* del filtro m. Usando la función lineal de *warping* por partes (3.1), $\hat{\omega}_m$ puede ser escrito como:

$$\hat{\omega}_m(\alpha) = \begin{cases} \alpha \cdot \omega_m & \omega_{\mathrm{m}} \le \omega_0 \\ \alpha \cdot \omega_0 + \dfrac{\omega_{\max} - \alpha \cdot \omega_o}{\omega_{\max} - \omega_0}(\omega_m - w_0) & \omega_{\mathrm{m}} > \omega_0 \end{cases} \tag{3.1}$$

donde $\omega_{\max}$ corresponde a la máxima frecuencia del banco de filtros, α es el parámetro o factor de *warping*, y ω_0 se define como en (2.21).

La energía del filtro m en la ventana i está indicada por $X_{i,m}$. El método VTLN propuesto en esta memoria estima la energía del filtro *warped* m, $\hat{X}_{i,m}$, como una combinación lineal de energías de filtros contiguos en el banco de filtros original: si el filtro *warped* m es cambiado hacia la izquierda ($\alpha \le 1$), la energía del filtro *warped* es estimada con una interpolación lineal entre $X_{i,m-1}$ y $X_{i,m}$; y, si el filtro *warped* m es cambiado hacia la derecha (i.e. $\alpha \le 1$), la energía del filtro *warped* es aproximada con una interpolación lineal entre $X_{i,m}$ y $X_{i,m+1}$. De acuerdo a eso, $\hat{X}_{i,m}$ se expresa como:

$$\hat{X}_{i,m}(\alpha) = \frac{X_{i,m} - X_{i,q}}{\omega_m - \omega_q} \cdot \left[\hat{\omega}_m(\alpha) - \omega_m^{ref} \right] + X_i^{ref} \tag{3.2}$$

donde

$$q = \begin{cases} m-1 & \alpha \le 1 \\ m+1 & \alpha > 1 \end{cases} \tag{3.3}$$

y, X_i^{ref} y ω_m^{ref} se definen como:

$$\hat{X}_{i,m}^{ref} = \frac{X_{i,m} + X_{i,q}}{2}$$
$$\omega_m^{ref} = \frac{\omega_m + \omega_q}{2} \tag{3.4}$$

VTLN convencional se implementa usualmente generando un banco de filtro por cada factor de *warping* a ser evaluado. Entonces, el α óptimo es aquel que provee la máxima verosimilitud. De acuerdo con el modelo presentado, las energías del banco de filtros por cada α evaluado pueden ser calculadas con (3.2) sin la necesidad de ejecutar un análisis de banco de filtros por cada α. Nótese que la interpolación lineal en (3.2) debiera ser una aproximación exacta para determinar $\hat{X}_{i,m}$ si $\omega_{m-1} \le \hat{\omega}_m$ o $\omega_{m+1} \ge \hat{\omega}_m$. Se observa empíricamente que estas condiciones son usualmente satisfactorias. Sino, en el peor caso, (3.2) se transforma en una extrapolación lineal y pierde exactitud, pero sigue siendo aplicable.

3.1.2 Búsqueda del factor de normalización óptimo

En vez de realizar una búsqueda por barrido, evaluando varios factores de *warping* para escoger el que maximice la verosimilitud, es posible estimar el α óptimo analíticamente. En esta sección se propone una optimización analítica de α basada en la estimación de máxima verosimilitud (MV). Aplicando la función de logaritmo natural a (3.2), la log-energía del filtro m puede escribirse como:

$$\log\left[\hat{X}_m(\alpha)\right] = \log\left(\frac{X_{i,m} - X_{i,q}}{\omega_m - \omega_q} \cdot \left[\hat{\omega}_m(\alpha) - \omega_m^{ref}\right] + X_{i,m}^{ref}\right) \tag{3.5}$$

Con objeto de simplificar la fórmula, sea $X_{i,m}^{L} = \log\left[X_{i,m} \right]$ y $\hat{X}_{i,m}^{L}(\alpha) = \log\left[\hat{X}_{i,m}(\alpha) \right]$. Al aplicar la aproximación por series de Taylor de primer orden al logaritmo en la función, de acuerdo a $\log(\alpha + \Delta\alpha) \cong \log(\alpha) + \frac{\Delta\alpha}{\alpha}$, cuando $\alpha \gg \Delta\alpha$, entonces $\hat{X}_{i,m}^{L}(\alpha)$ puede ser expresado como:

$$\hat{X}_{i,m}^{L}(\alpha) \cong \log\left(X_{i,m}^{ref} \right) + \frac{P_{i,m}}{X_{i,m}^{ref}} \left(\hat{\omega}_m(\alpha) - \omega_m^{ref} \right) = \left[\log\left(X_{i,m}^{ref} \right) - \frac{P_{i,m}}{X_{i,m}^{ref}} \omega_m^{ref} \right] + \frac{P_{i,m}}{X_{i,m}^{ref}} \hat{\omega}_m(\alpha) \quad (3.6)$$

donde $P_{i,m} = \left(X_{i,m} - X_{i,q} \right) / \left(\omega_m - \omega_q \right)$. Al definir $b_{i,m}^{1} = \left(1 / X_{i,m}^{ref} \right) P_{i,m}$ y $b_{i,m}^{0} = \log\left(X_{i,m}^{ref} \right) - b_{i,m}^{1} \cdot \omega_{i,m}^{ref}$ puede escribirse como:

$$\hat{X}_{i,m}^{L}(\alpha) \cong b_{i,m}^{1} \cdot \hat{\omega}_m(\alpha) + b_{i,m}^{0} \quad (3.7)$$

Observar que la aproximación de Taylor de primer orden requiere $X_{i,m}^{ref} \gg \frac{X_{i,m} - X_{i,q}}{\omega_m - \omega_q} \left[\hat{\omega}_m(\alpha) - \omega_m^{ref} \right]$. Al considerar $\left| \hat{\omega}_m(\alpha) - \omega_m^{ref} \right| < \left| \omega_m - \omega_q \right|$, la condición que satisface la serie de Taylor de primer orden puede ser evaluada por:

$$\left| \frac{X_{i,q} - X_{i,m}}{X_{i,m}^{ref}} \right| \leq \gamma \quad (3.8)$$

donde γ es un parámetro para descartar ventanas si la condición (3.8) no es satisfecha por los componenetes de $\hat{X}_{i,m}(\alpha)$, con $0 \leq m \leq M-1$. Al incorporar la definición de $\hat{\omega}_m(\alpha)$ en (3.7), de acuerdo a (3.1), $\hat{X}_{i,m}^{L}(\alpha)$ se puede reescribir como:

$$\hat{X}_{i,m}^{L}(\alpha) \gg \begin{cases} b_{i,m}^{1} \cdot \alpha \cdot \omega_m + b_{i,m}^{0} & \omega_m \leq \omega_0 \\ \alpha \cdot b_{i,m}^{1} \cdot \left[\omega_0 - \frac{\omega_o}{\omega_{\max} - \omega_0} (\omega_m - \omega_0) \right] + b_{i,m}^{1} \cdot \frac{\omega_{\max}}{\omega_{\max} - \omega_0} (\omega_m - \omega_0) + b_{i,m}^{0} & \omega_m > \omega_0 \end{cases} \quad (3.9)$$

Considerar que la secuencia de vectores de características MFCC observadas está denotado $X^C = \left\{ X_i^C \right\}_{i=0}^{I-1}$, donde: $X_i^C = \left\{ X_{i,n}^C \right\}_{n=0}^{N-1}$ corresponde a la ventana en el instante i, y I es el número total de ventanas de la señal; y $X_{i,n}^C$ denota el n^{vo} coeficiente cepstral de la ventana i, y N es el número total de coeficiente cepstrales estáticos. Luego, al aplicar la DCT, $X_{i,n}^C = \sum_{m=o}^{M-1} X_{i,m}^L \cos\left(\frac{\pi \cdot n}{M}(m-0.5) \right)$. Consecuentemente, al usar (3.9), $\hat{X}_{i,n}^C$ se puede escribir como:

$$\begin{aligned} \hat{X}_{i,n}^C(\alpha) &= \sum_{m=o}^{M-1} \hat{X}_{i,m}^L(\alpha) \cos\left(\frac{\pi \cdot n}{M}(m-0.5) \right) \\ &= \sum_{m=0}^{\substack{m_0 \\ \omega_{m_0} \le \omega_0}} \left(b_{i,m}^1 \cdot \alpha \cdot \omega_m + b_{i,m}^0 \right) \cos\left(\frac{\pi \cdot n}{M}(m-0.5) \right) \\ &\quad + \sum_{m=m_0+1}^{M-1} \left\{ \begin{array}{l} \alpha \cdot b_{i,m}^1 \left[\omega_0 - \frac{\omega_0}{\omega_{\max} - \omega_0}(\omega_m - \omega_0) \right] + \\ b_{i,m}^1 \frac{\omega_{\max}}{\omega_{\max} - \omega_0}(\omega_m - \omega_0) + b_{i,m}^0 \end{array} \right\} \cos\left(\frac{\pi \cdot n}{M}(m-0.5) \right) \end{aligned} \tag{3.10}$$

Observe que la ecuación lineal por partes que define $\hat{X}_{i,m}^L(\alpha)$ en (3.9) lleva a una representación DCT con dos sumatorias: la primera de $m=0$ a $m=m_o$, donde $\omega_{m_0} \le \omega_0$; y la segunda de $m=m_0+1$ a $m=M-1$. Si las sumas que dependen de α en (3.10) son separadas, $\hat{X}_{i,n}^C(\alpha)$ puede escribirse como:

$$\hat{X}_{i,n}^{C}(\alpha)=\alpha\cdot\left\{\begin{array}{l}\sum\limits_{\substack{m=0\\ \omega_{m_0}\le\omega_0}}^{m_0}\left(b_{i,m}^{1}\cdot\omega_m\right)\cos\left(\frac{\pi\cdot n}{M}(m-0.5)\right)\\ +\sum\limits_{m=m_0+1}^{M-1}\left\{b_{i,m}^{1}\left[\omega_0-\frac{\omega_0}{\omega_{\max}-\omega_0}(\omega_m-\omega_0)\right]\right\}\cos\left(\frac{\pi\cdot n}{M}(m-0.5)\right)\end{array}\right\}$$
$$+\sum_{\substack{m=0\\ \omega_{m_0}\le\omega_0}}^{m_0}b_{i,m}^{0}\cos\left(\frac{\pi\cdot n}{M}(m-0.5)\right)$$
$$+\sum_{m=m_0+1}^{M-1}\left\{b_{i,m}^{1}\frac{\omega_{\max}}{\omega_{\max}-\omega_0}(\omega_m-\omega_0)+b_{i,m}^{0}\right\}\cos\left(\frac{\pi\cdot n}{M}(m-0.5)\right) \quad (3.11)$$

Luego, definiendo

$$W_{i,n}=\sum_{\substack{m=0\\ \omega_{m_0}\le\omega_0}}^{m_0}\left(b_{i,m}^{1}\cdot\omega_m\right)\cos\left(\frac{\pi\cdot n}{M}(m-0.5)\right)$$
$$+\sum_{m=m_0+1}^{M-1}\left\{b_{i,m}^{1}\left[\omega_0-\frac{\omega_0}{\omega_{\max}-\omega_0}(\omega_m-\omega_0)\right]\right\}\cos\left(\frac{\pi\cdot n}{M}(m-0.5)\right) \quad (3.12)$$

y

$$B_{i,n}=\sum_{\substack{m=0\\ \omega_{m_0}\le\omega_0}}^{m_0}b_{i,m}^{0}\cos\left(\frac{\pi\cdot n}{M}(m-0.5)\right)$$
$$+\sum_{m=m_0+1}^{M-1}\left\{b_{i,m}^{1}\frac{\omega_{\max}}{\omega_{\max}-\omega_0}(\omega_m-\omega_0)+b_{i,m}^{0}\right\}\cos\left(\frac{\pi\cdot n}{M}(m-0.5)\right) \quad (3.13)$$

$\hat{X}_{i,n}^{C}(\alpha)$ puede expresarse como:

$$\hat{X}_{i,n}^{C}(\alpha)=\alpha\cdot W_{i,n}+B_{i,n} \quad (3.14)$$

3.1.2.1 Propuesta de algoritmo de VTLN

El algoritmo de frecuencia de warping propuesto en este paper hace uso del alineamiento de la mejor hipótesis de la primera decodificación proporcionado por el algoritmo de Viterbi. Considere que λ ha denotado una secuencia de fonemas HMMs dependientes de contexto, compuestos de K estados, donde s_k denota el estado número K dentro del HMM compuesto, con $1 \le k \le K$. $S = \left\{ s_{k(i)} \right\}_{i=0}^{I=1}$ también representa el alineamiento de la mejor hipótesis de la primera decodificación dado por el algoritmo de Viterbi calculado con X^C. S asocia cada ventana X_i en X^C con un estado dentro de λ denotado por $s_{k(i)}$. La aproximación presentada involucra tres pasos principales:

Paso 1: Dado una secuencia de vector de características X^C, el alineamiento de la mejor la hipótesis de la primera decodificación S *es* proporcionada por el algoritmo de Viterbi.

Paso 2: Empleando el modelo de frecuencia de *warping* basado en la interpolación de bancos de filtro propuesta aquí (IFE-VTLN), el parámetro optimo de *warping* α se obtiene por estimación ML haciendo uso de la mejor hipótesis de la primera decodificación del paso 1.

Paso 3: Finalmente, la secuencia de ventanas *warped* MFCC $\hat{X}^C$ se obtiene de acuerdo a (3.14).

En el paso 2, el parámetro de frecuencia de *warping* α se estima usarndo el criterio ML:

$$\hat{\alpha} = \arg\max_{\alpha} \left\{ p\left(X^C \middle| \lambda, S, \alpha \right) \right\} \tag{3.15}$$

donde $\hat{\alpha}$ es el parametro optimo de *warping* de frecuencia. Debido al hecho de que el modelo de energía de filtro interpolada dependa de ω_0 en (3.1) y (2.21), $\hat{\alpha}$ se estima asumiendo dos condiciones separadamente: $\hat{\alpha}_{izq}$ si $\alpha \le 1$; y $\hat{\alpha}_{der}$ si $\alpha > 1$. La estimación ML de $\hat{\alpha}$ se muestra en la fig. 2.

De acuerdo a la fig. 2, primero, $\hat{\alpha}_{izq}$ es calculada considerando $\alpha \leq 1$ y $\omega_0 = \frac{7}{8} \cdot \omega_{\max}$ como en (2.21). Luego, si $\alpha > 1$, se proponen dos iteraciones para estimar $\hat{\alpha}_{der}$:

Primero, con $\omega_0 = \frac{7}{8} \cdot \omega_{\max}$; segundo, con $\omega_0 = \frac{7}{8 \cdot \hat{\alpha}_{der}^{(1)}} \cdot \omega_{\max}$, donde $\hat{\alpha}_{der}^{(1)}$ es el factor *warping* óptimo obtenido en la iteración anterior. Finalmente, se escoge $\hat{\alpha}$ entre $\hat{\alpha}_{izq}$ y $\hat{\alpha}_{der}$ de acuerdo al que lleve a la máxima verosimilitud del alineamiento de la mejor hipótesis de la primera decodificación.

3.1.2.2 Estimación de máxima verosimilitud de α

Como resultado del alineamiento de la mejor hipótesis de la primera decodificación, se escoge la Gaussiano más probable por estado. Por consiguiente, el estado s_k se modela según una función Gaussiana con vector de medias $\mu_k = \{\mu_{k,n}\}_{n=0}^{N-1}$ y matriz diagonal de covarianzas Σ_k, y $\phi_k = (\mu_k, \Sigma_k)$. Las componentes de la diagonal de Σ_k son denotadas por $\sigma_k^2 = \{\sigma_{k,n}^2\}_{n=0}^{N-1}$. Luego, la probabilidad $p(X_i^C \mid \phi_{k(i)}, \alpha)$ se define como:

$$p(X_i^C \mid \phi_{k(i)}, A) = \frac{1}{(2\pi)^{\frac{N}{2}} \mid \Sigma_{k(i)} \mid^{\frac{1}{2}}} \cdot e^{-\frac{1}{2}\sum_{n=0}^{N-1} \frac{\left[\left(\alpha \cdot W_{i,n} + B_{i,n}\right) - \mu_{k(i),n}\right]^2}{\sigma_{k(i),n}^2}} \tag{3.16}$$

donde $\phi_{k(i)} = (\mu_{k(i)}, \Sigma_{k(i)})$ denota la serie de parámetros Gaussianos asociados al estado $s_{k(i)}$ asignado a la ventana X_i^C. El parámetro óptimo de frecuencia *warping* $\hat{\alpha}$ puede estimarse maximizando la probabilidad logarítmica de la siguiente función objetivo:

$$\hat{\alpha} = \arg\max_{\alpha} \left\{ \log\left[p\left(X^C \mid \lambda, S, \alpha \right) \right] \right\} = \arg\max_{\alpha} \left\{ \sum_{i=0}^{I-1} \log\left[p\left(X_i^C \mid \lambda, S, \alpha \right) \right] \right\} \tag{3.17}$$

Nótese que en la sumatoria sobre las ventanas de la señal, el índice de ventanas i debiera considerar solo aquellas ventanas cuyas energías de filtro cumplan la condición en (3.8). Luego, reemplazando (3.16) en (3.17), la optimización puede reescribirse como:

$$\hat{\alpha} = \arg\max_{\alpha} \left\{ \begin{array}{l} \sum\limits_{\substack{i=0 \\ |(X_{i,q}-X_{i,m})/X_{i,m}^{ref}|\le\gamma/\forall m}}^{I-1} \log\left[\left(\left(2\pi\right)^{\frac{N}{2}} |\,\Sigma_{k(i)}\,|^{\frac{1}{2}}\right)^{-1}\right] - \\ \\ \frac{1}{2}\cdot \sum\limits_{\substack{i=0 \\ |(X_{i,q}-X_{i,m})/X_{i,m}^{ref}|\le\gamma/\forall m}}^{I-1} \sum\limits_{n=0}^{N-1} \frac{\left[\alpha\cdot W_{i,n} + B_{i,n} - \mu_{k(i),n}\right]^2}{\sigma_{k(i),n}^2} \end{array} \right\} \tag{3.18}$$

donde $\sum\limits_{\substack{i=0 \\ |(X_{i,q}-X_{i,m})/X_{i,m}^{ref}|\le\gamma/\forall m}}^{I-1} \log\left[\left(\left(2\pi\right)^{\frac{N}{2}} |\,\Sigma_{k(i)}\,|^{\frac{1}{2}}\right)^{-1}\right]$ no depende de α y es descartado. Como resultado, α es estimado calculando la derivada parcial de (3.18) con respecto a α e igualando a cero:

$$\hat{\alpha} = \frac{\sum\limits_{\substack{i=0 \\ |(X_{i,q}-X_{i,m})/X_{i,m}^{ref}|\le\gamma/\forall m}}^{I-1} \sum\limits_{n=0}^{N-1} \frac{W_{i,n}\cdot\left(\mu_{k(i),n} - B_{i,n}\right)}{\sigma_{k(i),n}^2}}{\sum\limits_{\substack{i=0 \\ |(X_{i,q}-X_{i,m})/X_{i,m}^{ref}|\le\gamma/\forall m}}^{I-1} \sum\limits_{n=0}^{N-1} \frac{{W_{i,n}}^2}{\sigma_{k(i),n}^2}} \tag{3.19}$$

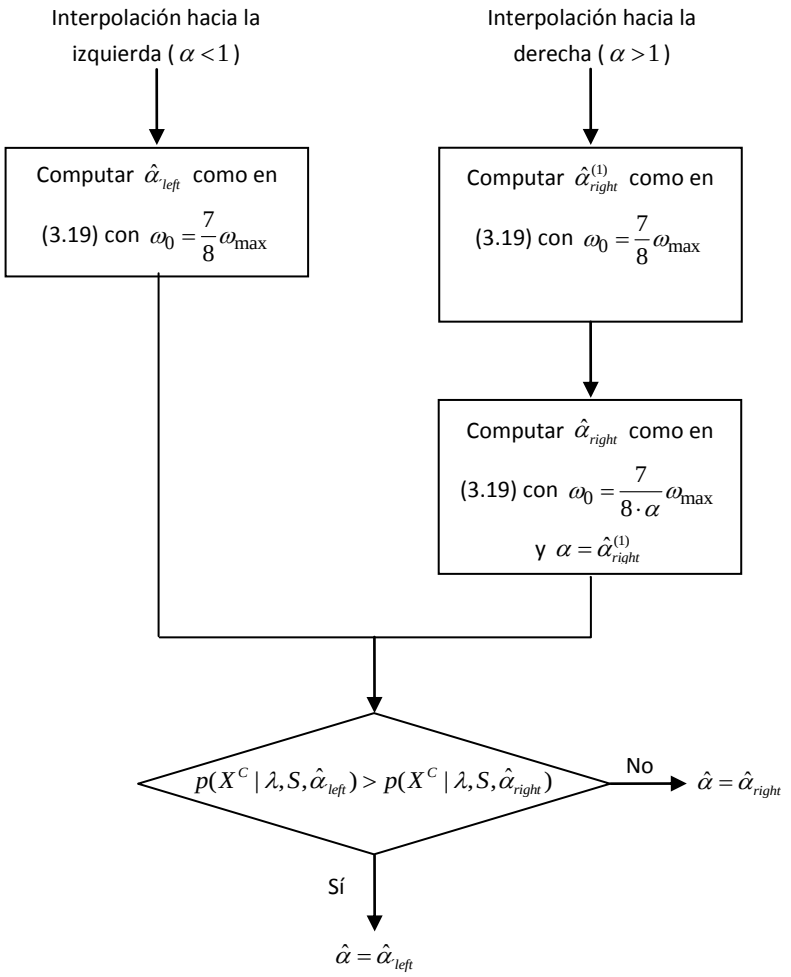


Figura 3.1. Diagrama de flujo del método propuesto de estimación analítica del factor de warping óptimo según el criterio de MV (IFE-VTLN –A).

3.2 Alineamiento de espectro mediante interpolaciones del vector de observación

En VTLN generalmente se modela la respuesta en frecuencia del tracto vocal como una función continua. Sin embargo, si se observa el espectro de la elocución de una misma palabra pronunciada por distintos locutores, es posible notar que la respuesta en frecuencia de cada locutor es única. Por ejemplo a veces algunas formantes tienen frecuencias más bajas que el modelo y otras formantes tienen frecuencias más altas. De aquí surge la motivación de normalizar

el vector de observación alineando el espectro en frecuencia de la elocución sin restringirlo a una determinada función de *warping*. En esta sección se presenta un modelo basado en esta motivación, explotando la idea de interpolación lineal de filtros vecinos usada para modelar VTLN en la sección 3.1.

3.2.1 Interpolación del vector de observación

Sea $X_{i,m}^{L}$ la energía en el dominio logarítmico del filtro m en el frame i. Sea $\hat{X}_{i,m}^{L}$ la energía normalizada del mismo filtro. Queremos expresar $\hat{X}_{i,m}^{L}$ una combinación lineal de energías de filtro contiguas en el banco de filtros original. De acuerdo a esto, $\hat{X}_{i,m}^{L}$ es expresado como:

$$\hat{X}_{i,m}^{L} = X_{i,m}^{L} + \left(X_{i,m-1}^{L} - X_{i,m}^{L}\right)\cdot\alpha_{m} + \left(X_{i,m+1}^{L} - X_{i,m}^{L}\right)\cdot\alpha_{m+M}$$

Definiendo $\Delta_{i,m} = X_{i,m-1}^{L} - X_{i,m}^{L}$ y $\Delta_{i,m+M} = X_{i,m+1}^{L} - X_{i,m}^{L}$, $\hat{X}_{i,m}^{L}$ se puede expresar como:

$$\hat{X}_{i,m}^{L} = X_{i,m}^{L} + \Delta_{i,m}\cdot\alpha_{m} + \Delta_{i,m+M}\cdot\alpha_{m+M} \tag{3.20}$$

Bajo la restricción:

$$\left(\alpha_{m} = 0\right) \vee \left(\alpha_{m+M} = 0\right) \tag{3.21}$$

A diferencia del método propuesto en 3.1, donde el parámetro que se estima es un único α, en el método que se formula en esta sección cada filtro se interpola linealmente con los filtros vecinos de acuerdo a 2 variables, α_{m} y α_{m+M}, totalizando un total de $2*M$ variables o grados de libertad, donde M es el número de filtros usados en la parametrización. En los experimentos se usa un banco de 14 filtros, por lo tanto son 28 variables. La condición (3.22) busca forzar que α_{m} o α_{m+M} sea igual a cero. Esta condición es necesaria para que $\hat{X}_{i,m}^{L}$ sea una interpolación lineal entre filtros vecinos hacia un único lado, sino se estaría sumando un porcentaje de la diferencia entre filtros vecinos sin que tenga mucho sentido en cuanto a lo que se busca modelar.

Para los casos extremos $\hat{X}_{i,1}^{L}$ y $\hat{X}_{i,M}^{L}$ no se tiene un filtro a la izquierda y la derecha respectivamente para poder interpolar. En estos casos solamente se considera una variable.

Dado que cada filtro se modifica de acuerdo a un parámetro único, se está buscando alinear el espectro (IFE-SA, *Interpolation of Filter Energies – Spectral Alignment*) de la elocución con el alineamiento de la mejor hipótesis de la primera codificación

3.2.2 Búsqueda del vector de normalización óptimo

Dado el alto número de variables a optimizar para la fórmula propuesta, y la no independencia entre ellas realizar una búsqueda exhaustiva de grilla resulta infactible computacionalmente, dado que el número de combinaciones a evaluar sería del orden de D^{28}, donde D es el número de valores que se puede asignar a cada variable (discretización). Sin embargo, al obviar la condición (3.22) es posible encontrar analíticamente el vector óptimo $\vec{\alpha}$ maximizando la ecuación (3.17).

Es posible expresar $\hat{X}_{i,n}^{C}$ como:

$$\begin{aligned}\hat{X}_{i,n}^{C} &= \sum_{m=1}^{M}\left\{\hat{X}_{i,m}^{L}\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot\left(m-0.5\right)\right)\right\} \\ &= \sum_{m=1}^{M}\left\{\left(X_{i,m}^{L}+\Delta_{i,\alpha,m}\cdot\alpha_m+\Delta_{i,\gamma,m}\cdot\gamma_m\right)\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot\left(m-0.5\right)\right)\right\}\end{aligned} \tag{3.22}$$

Si reemplazamos (3.14) por (3.22) en la sección (3.1) la optimización (3.18) puede reescribirse como:

$$\hat{\alpha}=\arg\max_{\alpha}\left\{\begin{array}{c}\displaystyle\sum_{i=0}^{I-1}\log\left[\left(\left(2\pi\right)^{\frac{N}{2}}\left|\Sigma_{k(i)}\right|^{\frac{1}{2}}\right)^{-1}\right]- \\ \displaystyle\frac{1}{2}\cdot\sum_{i=0}^{I-1}\sum_{n=0}^{N-1}\frac{\left[\displaystyle\sum_{m=1}^{M}\left\{\left(\begin{array}{l}X_{i,m}^{L}+\Delta_{i,m}\cdot\alpha_m+\\ \Delta_{i,m+M}\cdot\alpha_{m+M}\end{array}\right)\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot\left(m-0.5\right)\right)\right\}-\mu_{i,n}\right]^2}{\sigma_{i,n}^2}\end{array}\right\} \tag{3.23}$$

donde $\sum_{i=0}^{I-1}\log\left[\left(\left(2\pi\right)^{\frac{N}{2}}\left|\Sigma_{k(i)}\right|^{\frac{1}{2}}\right)^{-1}\right]$ no depende de $\vec{\alpha}$ y es descartado.

Desarrondo la sumatoria dependiente de $\vec{\alpha}$ en (3.23) como un cuadrado de binomio y agrupando términos:

$$-\frac{1}{2}\cdot\sum_{i=0}^{I-1}\sum_{n=0}^{N-1}\frac{\left[\sum_{m=1}^{M}\left\{\left(X_{i,m}^{L}+\Delta_{i,m}\cdot\alpha_m+\Delta_{i,m+M}\cdot\alpha_{m+M}\right)\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot\left(m-0.5\right)\right)\right\}-\mu_{i,n}\right]^2}{\sigma_{i,n}^2}$$

$$=-\left\{\frac{1}{2}\cdot\sum_{m=1}^{M}\sum_{p=1}^{M}\alpha_m\cdot\alpha_p\cdot\sum_{i=0}^{I-1}\sum_{n=0}^{N-1}\frac{\left[\Delta_{i,m}\cdot\Delta_{i,p}\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot\left(m-0.5\right)\right)\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot\left(p-0.5\right)\right)\right]}{\sigma_{i,n}^2}\right.$$

$$+\sum_{m=1}^{M}\sum_{p=1}^{M}\alpha_m\cdot\alpha_{p+M}\cdot\sum_{i=0}^{I-1}\sum_{n=0}^{N-1}\frac{\left[\Delta_{i,m}\cdot\Delta_{i,p+M}\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot\left(m-0.5\right)\right)\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot\left(p-0.5\right)\right)\right]}{\sigma_{i,n}^2}$$

$$+\frac{1}{2}\cdot\sum_{m=1}^{M}\sum_{p=1}^{M}\alpha_{m+M}\cdot\alpha_{p+M}\cdot\sum_{i=0}^{I-1}\sum_{n=0}^{N-1}\frac{\left[\Delta_{i,m+M}\cdot\Delta_{i,p+M}\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot\left(m-0.5\right)\right)\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot\left(p-0.5\right)\right)\right]}{\sigma_{i,n}^2}$$

$$+\sum_{m=1}^{M}\alpha_m\cdot\sum_{i=0}^{I-1}\sum_{n=0}^{N-1}\frac{\left[\Delta_{i,m}\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot\left(m-0.5\right)\right)\cdot\left(\sum_{p=1}^{M}\left\{X_{i,p}^{L}\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot\left(p-0.5\right)\right)\right\}-\mu_{i,n}\right)\right]}{\sigma_{i,n}^2}$$

$$+\sum_{m=1}^{M}\alpha_{m+M}\cdot\sum_{i=0}^{I-1}\sum_{n=0}^{N-1}\frac{\left[\Delta_{i,m+M}\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot\left(m-0.5\right)\right)\cdot\left(\sum_{p=1}^{M}\left\{X_{i,p}^{L}\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot\left(p-0.5\right)\right)\right\}-\mu_{i,n}\right)\right]}{\sigma_{i,n}^2}$$

$$+\frac{1}{2}\cdot\sum_{m=1}^{M}\sum_{p=1}^{M}\sum_{i=0}^{I-1}\sum_{n=0}^{N-1}\frac{\left[X_{i,m}^{L}\cdot X_{i,p}^{L}\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot\left(m-0.5\right)\right)\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot\left(p-0.5\right)\right)\right]}{\sigma_{i,n}^2}$$

$$\left.-\sum_{i=0}^{I-1}\sum_{n=0}^{N-1}\frac{\mu_{i,n}\cdot\sum_{m=1}^{M}\left\{X_{i,m}^{L}\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot\left(m-0.5\right)\right)\right\}}{\sigma_{i,n}^2}+\frac{1}{2}\cdot\sum_{i=0}^{I-1}\sum_{n=0}^{N-1}\frac{\mu_{i,n}^2}{\sigma_{i,n}^2}\right\}$$

Escribiendo en forma matricial la ecuación anterior, se obtiene el "Puntaje de versosimilitud".

$$Puntaje\ de\ verosimilitud = \frac{1}{2}\cdot\vec{\alpha}^{t}\cdot A\cdot\vec{\alpha}+\vec{\alpha}^{t}\cdot\vec{P}+K \tag{3.24}$$

siendo,

$$A_{m,p} = -\sum_{i=0}^{I-1}\Delta_{i,m}\cdot\Delta_{i,p}\sum_{n=0}^{N-1}\frac{\left[\cos\left(\frac{\pi\cdot n}{M}\cdot(m-0.5)\right)\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot(p-0.5)\right)\right]}{\sigma_{i,n}^{2}},$$

$$A_{m+M,p} = A_{m,p+M} = -\sum_{i=0}^{I-1}\Delta_{i,m}\cdot\Delta_{i,p+M}\sum_{n=0}^{N-1}\frac{\left[\cos\left(\frac{\pi\cdot n}{M}\cdot(m-0.5)\right)\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot(p-0.5)\right)\right]}{\sigma_{i,n}^{2}},$$

$$A_{m+M,p+M} = -\sum_{i=0}^{I-1}\Delta_{i,m+M}\cdot\Delta_{i,p+M}\sum_{n=0}^{N-1}\frac{\left[\cos\left(\frac{\pi\cdot n}{M}\cdot(m-0.5)\right)\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot(p-0.5)\right)\right]}{\sigma_{i,n}^{2}},$$

$$P_{p} = -\sum_{i=0}^{I-1}\sum_{n=0}^{N-1}\frac{\left[\sum_{m=1}^{M}\left\{X_{i,m}^{L}\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot(m-0.5)\right)\right\}-\mu_{i,n}\right]\cdot\left[\Delta_{i,p}\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot(p-0.5)\right)\right]}{\sigma_{i,n}^{2}},$$

$$P_{p+M} = -\sum_{i=0}^{I-1}\sum_{n=0}^{N-1}\frac{\left[\sum_{m=1}^{M}\left\{X_{i,m}^{L}\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot(m-0.5)\right)\right\}-\mu_{i,n}\right]\cdot\left[\Delta_{i,p+M}\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot(p-0.5)\right)\right]}{\sigma_{i,n}^{2}}$$

$$K = \begin{cases} -\frac{1}{2}\cdot\sum_{m=1}^{M}\sum_{p=1}^{M}\sum_{i=0}^{I-1}\sum_{n=0}^{N-1}\frac{\left[X_{i,m}^{L}\cdot X_{i,p}^{L}\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot(m-0.5)\right)\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot(p-0.5)\right)\right]}{\sigma_{i,n}^{2}} \\ +\sum_{i=0}^{I-1}\sum_{n=0}^{N-1}\frac{\mu_{i,n}\cdot\sum_{m=1}^{M}\left\{X_{i,m}^{L}\cdot\cos\left(\frac{\pi\cdot n}{M}\cdot(m-0.5)\right)\right\}}{\sigma_{i,n}^{2}}-\frac{1}{2}\cdot\sum_{i=0}^{I-1}\sum_{n=0}^{N-1}\frac{\mu_{i,n}^{2}}{\sigma_{i,n}^{2}} \end{cases}$$

$\forall m, p = 1,2,\ldots,M$

Para encontrar el vector de parámetros óptimo $\vec{\alpha}$ se calcula la derivada parcial de (3.24) con respecto a cada variable de $\vec{\alpha}$ e igualando a cero. Es fácil ver que el óptimo de la ecuación (3.24) se encuentra al resolver el siguiente sistema:

$$A \cdot \vec{\alpha} + \vec{P} = 0$$

Por lo tanto,

$$\vec{\alpha}_{opt} = -A^{-1} \cdot \vec{P} \tag{3.25}$$

Se incluye una restricción al calcular $\vec{\alpha}_{opt}$ según (3.25):

$$-0.1 \le \alpha_m \le 1.1 \quad \forall m = 1,2,...,2 \cdot M \tag{3.26}$$

En general α_m debería estar entre 0 y 1, porque fuera de este rango implicaría que se está extrapolando. En caso de que para algún m, α_m no cumpla (3.26), α_m se hace igual a la cota más cercana.

Al resolver (3.25) se está considerando el modelo definido en (3.20) sin la condición (3.22). Incluir en una solución analítica la condición (3.22) es imposible. Para encontrar el óptimo sería necesario realizar la optimización forzando que siempre se cumpla (3.22), lo que implicaría realizar la optimización para $2^{2 \cdot M} = 2^{28}$ (aproximadamente 268 millones de combinaciones) casos distintos. Hacer esto es impracticable, por eso se soluciona este problema optimizando localmente con una búsqueda codiciosa, provocando que la solución final pueda o no ser el óptimo global. La solución consta de los siguientes pasos:

Paso 1: Calcular la matriz y vector de (3.24). A esta matriz y vector los llameremos globales.

Paso 2: Considerar nulas todas las interpolaciones hacia la derecha. Para esto se crea una matriz A' y vector $\vec{P}'$ igualando a cero todas las filas y columnas que incorporen en su cálculo valores de las interpolaciones que se consideran nulas, exceptuando las componentes diagonales de la matriz A' que se hacen igual a 1 para que la matriz tenga inversa (sino el determinante es 0). Calcular $\vec{\alpha}$ de acuerdo a (3.25). Calcular el "Puntaje de verosimilitud" usando (3.24).

Paso 3: Idem paso 2, pero considerando nulas todas las interpolaciones hacia la izquierda. Crear la matriz A' y vector $\vec{P}'$, para luego obtener el vector óptimo y finalmente calcular el "Puntaje de verosimiltud".

Paso 4: Elegir entre la matriz y vector del Paso 2 y Paso 3 como matriz y vector iniciales A_0 y $\vec{P}_0$, siendo el criterio de elección el máximo "Puntaje de verosimiltud".

Paso 5: Permutar la interpolación del segundo filtro (la del primero no cambia puesto que solamente interpola hacia la derecha). Esto es, si se está considerando interpolación hacia la derecha, considerar ahora interpolación hacia la izquierda y viceversa. Para ello se deben hacen los cambios respectivos en la matriz A_0 y vector $\vec{P}_0$ y luego recalcular $\vec{\alpha}$. Calcular el "Puntaje de verosimiltud" para este caso. Guardar A_0 y $\vec{P}_0$ y usar como puntaje máximo este nuevo valor en caso de que el puntaje calculado supere el puntaje máximo. Retornar A_0 y $\vec{P}_0$ a sus valores originales.
Repetir lo anterior con los demás filtros. Al terminar cambiar a A_0 y $\vec{P}_0$ de acuerdo al puntaje máximo obtenido.

Paso 6: Repetir Paso 5 con nuevos valores de A_0 y $\vec{P}_0$ hasta que no se modifiquen más. Obtener $\vec{\alpha}_{óptimo}$ con A_0 y $\vec{P}_0$ y calcular los coeficientes cepstrales normalizados con $\vec{\alpha}_{óptimo}$.

El modelo propuesto también podría aplicarse en el dominio lineal de las energías y luego usar la aproximación usada en (3.6) para realizar la optimización. Sin embargo debido al error que se incorpora con la aproximación lineal del logaritmo, el puntaje de verosimilitud no sería exacto, trasladándose este error a la optimización iterativa usada para elegir el lado de interpolación (no implica que los resultados serían peores, pero en esta memoria no se experimenta con esta variante).

4 Experimentos con técnicas de normalización en ASR

4.1 Experimentos con VTLN estándar en ASR

4.1.1 Transformación del banco de filtros

La transformación del banco de filtros se hace aplicando la función de *warping* expresada en (2.20) sobre la frecuencia mínima, central y máxima de cada filtro, como se explicó en **Error! Reference source not found.**. Luego se generan las rectas en el dominio Mel de manera que los filtros conserven su forma triangular, aunque asimétrica para $\alpha \neq 1$.

En la **Figura 2.4** se ve un ejemplo, del banco de filtros Mel original y uno Mel *warped* de acuerdo a la transformación explicada.

4.1.2 Búsqueda del factor de normalización óptimo

Para buscar el factor de normalización óptimo se realiza una búsqueda en barrido, evaluando valores de α entre 0.85 y 1.15 con espaciado de 0.01, totalizando 31 valores para α.

4.1.3 Reconocimiento utilizando factor de normalización óptimo

	Baseline	Standard VTLN
WER(%)	4.15	3.64

Tabla 4.1. WER(%) obtenido con el sistema base y VTLN estándar.

4.2 Experimentos con IFE-VTLN en ASR

4.2.1 Reconocimiento utilizando factor de normalización óptimo

	Baseline	Standard VTLN	IFE-VTLN-G
WER(%)	4.15	3.64	2.23

Tabla 4.2. WER(%) obtenido con el sistema base, VTLN estándar e IFE-VTLN-G.

	γ					
	0.5	0.6	0.7	0.8	0.9	1.0
WER(%)	2.50	2.53	2.50	2.50	2.53	2.50
% de frames considerados	2.08	8.49	22.83	47.03	79.51	100.00

Tabla 4.3. WER(%) con IFE-VTLN-A y porcentaje de *frames* que satisfacen la condición (3.8) vs. γ, como se define en (3.8).

4.3 Experimentos con IFE-SA en ASR

4.3.1 Reconocimiento utilizando factor de normalización óptimo

	Baseline	Standard VTLN	IFE-SA
WER(%)	4.15	3.64	3.40

Tabla 4.4. WER(%) obtenido con el sistema base, VTLN estándar e IFE-SA.

4.4 Condiciones de evaluación

4.4.1 Experimentos con base de datos en ambiente limpio LATINO-40

Los resultados de reconocimiento de voz continuo y de locutor independiente presentados en esta memoria fueron obtenidos usando una tarea de vocabulario medio grabado en un entorno limpio, la base de datos LATINO-40 (Bernstein et. al, 1995). Esta base de datos esta compuesta por grabaciones de habla continuas de 40 locutores nativos de latinoamerica, con cada locutor leyendo 125 oraciones de periódicos en español. Los datos de entrenamiento corresponden a 4500 oraciones proporcionadas por 36 locutores. El vocabulario esta compuesto por casi 6000 palabras. La base de datos de ensayo contiene 500 elocuciones (4000 palabras) proporcionadas por cuatro locutores de prueba (dos femeninos y dos masculinos) que no son parte del grupo de locutores usados en el entrenamiento. Cada elocución tiene una duración promedio de 4.6 segundos, y el material de entrenamiento y ensayo corresponde a 5.8 horas y 0.6 horas de grabación, respectivamente.

Las señales de habla se dividieron en ventanas de 25 ms con cincuenta porciento de superposición. La banda de 300 a 3400 Hz fue cubierta por 14 filtros Mel DFT, y en la salida de cada canal se calculó el logaritmo de la energía. Treina y tres parametros MFCC (coeficientes estatico, delta y delta-delta) fueron calculados por cada ventana. La técnica de *Cepstral Mean Normalization* (CMN) también fue empleada. La oración reconocida correspondía a la primera hipótesis (la más probable) dentro de la lista la *N-best* obtenida de la decodificación de Viterbi. Cada trifonema fue modelado con una topología de tres estados de izquierda a derecha sin

saltarse estados, con distribuciones multivariantes de ocho Gaussianas por cada estado con matrices de covarianza diagonales. Los HMMs fueron entrenados usando HTK y se empleó un modelo de lenguaje de trigramas durante el reconocimiento. La decodificación de Viterbi se obtiene con un motor de reconocimiento implementado en el Laboratorio de Procesamiento y Transmisión de Voz de la Universidad de Chile. El sistema base dio un WER igual a 4.15%. El modelo de energía de filtro interpolada propuesto es aplicado por medio de una búsqueda en barrido utilizando el criterio de MV, IFE-VTLN-G, y la propuesta de estimación analítica basada en el criterio de MV, IFE-VTLN-A.

4.5 Resultados y discusión

En esta sección comparan y discuten los resultados obtenidos en los experimentos con los métodos propuestos en el Capítulo 3. Para comparar el desempeño de las técnicas propuestas se realizaron experimentos con una técnica convencional de VTLN. Como puede verse en la **Tabla 4.2**, la búsqueda en barrido basada en el criterio de MV con VTLN convencional, proporciona una reducción en WER igual a 12.3% comparada con el sistema base. Esta mejora esta dentro de un rango esperable, si se toman en cuenta los resultados expuestos en la literatura especializada.

4.5.1 Evaluación del rendimiento con IFE-VTLN

El esquema IFE-VTLN-G propuesto lleva a redudicciones de WER tan altos como 46.3% y 38.7% si se compara con el sistema base y VTLN convencional respectivamente. El resultado corrobora la hipótesis considerada aquí sobre las perturbaciones introducidas en la estimación de energía de filtro debido a discontinuidades causadas por DFT y la estructura armónica de señales sonoras cuando la frecuencia central del filtro de pasa banda es modificada.

La **Tabla 4.3** presenta los resultados proporcionados por la estimación analítica basada en el criterio ML del factor de *warping* óptimo acorde a la **Figura 3.1**, IFE-VTLN-A. Como puede verse en la **Tabla 4.3**, IFE-VTLN-A proporciona reducciones en WER tan altas como 39.8% y 31.3% si se compara con el sistema base y con VTLN convencional, respectivamente. Este resultado sugiere que la restricción requerida por la expansión de Taylor de primer orden para la función logarítimica se cumple generalmente. Además, observe que la mejora debido a IFE-

VTLN-A dificilmente depende de γ. Por consecuencia, el factor de *warping* óptimo podría estimarse con un bajo porcentaje del total de ventanas en la elocución.

4.5.2 Evaluación del rendimiento con IFE-SA

En la **Tabla 4.4** se muestran los resultados obtenidos con el esquema IFE-SA, que lleva a reducciones de WER de un 18.1% y 6.1% si se compara con el sistema base y el VTLN convencional respectivamente.

IFE-VTLN-A obtiene una reducción de WER de un 26.5% si se compara con IFE-SA. Este es un porcentaje alto de mejora para sistemas reconocimiento de voz. Sin embargo, es importante destacar que los experimentos se realizaron con una base de datos de 4 locutores. Esta no es una cantidad suficientemente grande de locutores como para ser concluyentes con respecto a la superioredad de una técnica sobre la otra. Con IFE-VTLN es posible estimar el factor de *warping* óptimo con un porcentaje bajo de *frames* (menor a 10%). Si bien esta característica de la técnica puede ser positiva, también es una limitante, desde el punto de vista que la normalización con esta técnica no debería variar mucho al usar varias señales de adaptación. Por otro lado la técnica IFE-SA tiene un alto potencial de mejora usando más señales de adaptación debido al alto número de parámetros que deben estimarse.

4.6 Conclusiones

En esta memoria se propone un método de VTLN modelando el *warping* en frecuencia como la interpolación lineal de energías de filtro contiguas. Además de presenta un método de optimización analítica basada en el criterio de máxima verosimilitud para estimar el factor de *warping.* Experimentos con ejercicios continuos de reconocimiento vocal con LATINO-40 de vocabulario medio, demuestran que el modelo de interpolación de energías de los filtros con búsqueda por barrido basada en el criterio de MV puede llevar a reducciones de WER tan altas como 46.3% y 38.7% si se compara con el sistema base y el VTLN normal respectivamente. Además, el esquema analítico de optimización presentado, alcanza reducciones de WER iguales a 39.8% y 31.3% si se compara con el sistema base y el VTLN normal, respectivamente.

Se propone un segundo método IFE-SA, que posee más grados de libertad que IFE-VTLN. Esta técnica busca alinear el espectro de la señal mediante interpolaciones lineales de energías de filtros contiguas, existiendo dos parámetros optimizables por cada filtro usado en la parametrización. El esquema IFE-SA propuesto lleva a reducciones de WER de un 18.1% y 6.1% si se compara con el sistema base y el VTLN normal respectivamente.

5 Conclusiones y Propuestas para Trabajo Futuro

5.1 Conclusiones

Se han propuesto he implementado dos métodos de normalización de parámetros con el objetivo de mejorar la robustez a la variabilidad de locutor en ASR. Ambos métodos se basan en la interpolación lineal de energías de filtros vecinos.

En el método IFE-VTLN se modela el *warping* en frecuencia como una interpolación lineal de filtros contiguos. Para buscar el parámetro o factor de *warping* óptimo se maximiza la verosimilitud del vector de parámetros normalizados con respecto al alineamiento de la mejor hipótesis de la primera decodificación proporcionado por el algoritmo de Viterbi. Esta optimización se puede resolver con una búsqueda en barrido o analíticamente bajo ciertas suposiciones y aproximaciones. Se obtienen reducciones en el WER de 46.3 y 38.7% si se comparan con el sistema base y VTLN convencional, respectivamente. Además, el esquema analítico de optimización presentado, alcanza reducciones de WER iguales a 39.8% y 31.3% si se compara con el sistema base y VTLN convencional, respectivamente. La búsqueda en barrido es aproximadamente dos veces más rápida que VTLN convencional (que también consiste en un barrido), mientras que la optimización analítica es aproximadamente 16 veces más rápida.

En el método propuesto IFE-SA se usa la interpolación lineal de filtros contiguos de forma similar a IFE-VTLN, pero cada filtro se modifica de forma independiente. Esto significa que existe un parámetro que determina el nivel de interpolación por cada filtro y no un único parámetro que modifica todos los filtros como en IFE-VTLN. Para encontrar el vector de parámetros óptimo se debe realizar una búsqueda analítica, puesto que una búsqueda en barrido es imposible de realizar en la práctica. Al buscar analíticamente la solución se obtienen disminuciones en el WER de un 18.1% y 6.1% cuando se compara con el sistema base y VTLN convencional respectivamente. El tiempo de cómputo que requiere este esquema de normalización es aproximadamente 5 veces más rápida que VTLN convencional.

En base a los resultados y a los objetivos de esta memoria es posible concluir entonces que los métodos propuestos mejoran la robuztez a la variabilidad de locutor en los sistemas de ASR. Estas técnicas tienen además la caracterísica de poder ser resueltas analíticamente. Ambas son computacionalmente más eficientes que VTLN convencional. Se cumplen entonces los objetivos enunciados en el comienzo de esta memoria.

5.2 Propuestas para trabajo futuro

En esta sección final de la memoria se presenta una lista de propuestas para trabajo futuro, que incluyen ideas que podrían ser útiles para quien quisiera continuar perfeccionando los métodos propuestos.

1) Entrenar modelos normalizados con VTLN (válido tanto para modelo de 1 variable como de múltiples variables). Una vez que se tiene el modelo entrenado, se deben cargar el modelo general y el modelo normalizado en el reconocedor, hacer el primer alineamiento con el modelo general y luego aplicar VTLN. Para aplicar VTLN hay al menos dos opciones que pueden considerarse: **a)** hacer VTLN con respecto al modelo general; **b)** hacerlo considerando (forzadamente) el modelo normalizado. Luego se hace una nueva decodificación (Viterbi) con el modelo normalizado. La motivación de normalizar el modelo, es que dado los métodos usados buscan normalizar el locutor, las variaciones inter-locutor deberían reducirse (varianzas del modelo deberían ser menores), pudiendo mejorarse la tasa de aciertos del sistema. Notar que si se usa **b)** como solución el factor de

normalización va a depender del modelo normalizado que se tenga, por lo que podría reentrenarse el modelo normalizado sucesivas veces.

2) Calcular las energías de los filtros para 3 casos: sin *warping*, con *warping* extremo hacia la izquierda y con *warping* extremo hacia la derecha. Con esto, es posible hacer *warping* de forma similar a IFE-VTLN, pero interpolando los filtros con las versiones *warped* de sí mismos, en vez de realizar la interpolación con filtros vecinos. Además los factores de *warping* extremos hacia la derecha e izquierda pueden ser parámetros únicos para cada filtro. Esto agrega grados de libertad al algoritmo, siendo el modelo de *warping* de 3.1.1 un caso particular de este método (i.e. para un cierto factor de *warping*, la energía del filtro n *warped* es igual al filtro $n-1$).
3) Generar una rutina que haga un realineamiento, es decir un alineamiento forzado a una secuencia de estados ya generado por una decodificación anterior (i.e. Viterbi). Esta rutina podría usarse por ejemplo para hacer un realineamiento de una señal normalizada sobre cada alineamiento de la estructura *N-best* generada con el primer alineamiento de Viterbi y obtener el puntaje de verosimilitud. Así, es posible quedarse con el reconocimiento que maximiza la verosimilitud de este realineamiento, sin que sea necesario hacer una segunda decodificación con el algoritmo de Viterbi. De esta forma se hace más rápida la etapa de normalización. La motivación es que en general se espera que la transcripción correcta esté en alguna de las estructuras encontradas por la primera decodificación con el algoritmo de Viterbi, por lo que al usar este método se deberían obtener prácticamente los mismo resultados que haciendo una segunda decodificación. También podría usarse para la técnica b) propuesta en 1), realineando la secuencia de estados de acuerdo al modelo normalizado, previo a la búsqueda de los parámetros de normalización.
4) Aplicar las técnicas de normalización propuestas iterativamente. Las técnicas propuestas se aplican sobre las energías de los filtros, para obtener un nuevo valor de energía para cada filtro. Es posible aplicar el mismo proceso de normalización sobre las energías ya normalizadas. Este proceso se puede aplicar sucesivas veces, normalizando con la nueva señal de la misma forma que la primera normalización. También se puede desarrollar una técnica que use una combinación de los valores originales de las energías con los valores normalizados. La cantidad de iteraciones a realizar sería otro parámetro ajustable que

debe elegirse cuidando no sobreajustar los parámetros al alineamiento de la mejor hipótesis del reconocimiento preliminar.

5) Obtener los parámetros de normalización para cada alineamiento de la estructura *N-best*. Estos parámetros podrían ser útiles para observar por ejemplo en qué zona se concentran y tomar una decisión con respecto al lado que se elige. También se podría realizar la normalización para cada caso y luego recalcular la verosimilitud usando la rutina de realineamiento descrita en 3). Finalmente podría tomarse como factor óptimo aquel con el que logra la mayor verosimilitud.
6) Combinar las técnicas de optimización analítica con ruido aditivo y técnicas de compensación de canal. Si las técnicas de compensación dependen del valor de la energía de cada filtro, se puede resolver el problema iterativamente: en cada iteración se normalizan los vectores de observación considerando ruido y/o compensación de canal para el valor inicial (en la iteración) de la energía de los filtros. Como los cambios de los vectores de observación al normalizar son pequeños, el ruido y/o compensación de canal variará en una proporción pequeña a su valor inicial, por lo que el resultado óptimo de la optimización debiera converger al iterar.
7) Incluir una componente constante (sesgo) en la normalización que se realiza con IFE-SA. Esto es sumar una constante a cada filtro o bien a cada componente cepstral (no es lo mismo, puesto que son más filtros que coeficiente cepstrales). Esto implicaría un mayor número de parámetros y por lo tanto mayor extensión del sistema de ecuaciones a resolver.
8) En el sistema de reconocimiento usado en el laboratorio se usan varias técnicas de compensación adicionales a las expuestas en esta memoria, con el objetivo de mejorar el reconocimiento. Estas técnicas no están explicitadas en esta memoria, sin embargo dado que al normalizar la señal con cualquier técnica se busca maximizar la verosimilitud con el alineamiento de la mejor hipótesis de la primera decodificación, al incluir las técnicas posteriores se la aleja la normalización usada de la normalización óptima real (la medida de verosimilitud usada es una aproximación). Además es posible que parte de lo que se quiere lograr con estas técnicas sea compensado con la normalización por sí sola, sobre

todo con la normalización con múltiples variables. Sino, se deberían incluir de alguna forma las técnicas dentro de la optimización como se explica en el punto 6).

9) Se observaron similitudes en los vectores de normalización obtenidos mediante IFE-SA para elocuciones distintas de un mismo locutor. Esto lleva a pensar que sería interesante la aplicación de esta técnica en sistema de identificación de locutor. Para esto, podría guardarse el vector de normalización durante el entrenamiento. Luego se podría usar este vector para normalizar cualquier señal que intente identificarse como aquel locutor. La motivación es que podría aumentarse la variabilidad en el universo de locutores, aumentando en consecuencia la tasa de aciertos del sistema.
10) Usar *codebooks* para la técnica de alineamiento de espectro con múltiples variables propuesta en 3.2 en vez de usar el alineamiento de la mejor hipótesis de la decodificación (idea aportada por C. Garretón). De este modo no se requiere de una primera decodificación.

6 Referencias

Acero, A., 1990. Acoustical and Environmental Robustness in Automatic Speech Recognition. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, Sept. 1990.

Acero, A. and Stern, R.M., 1991. Robust Speech Recognition by Normalization of the Acoustic Space. Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Vol. 2, pp. 893–896, Toronto, Canada, May 1991.

Becchetti, C., and Prina, L., 1999. Speech recognition, theory and c++ implementation. Wiley E. London, UK.

Bernstein, J., 1995. et al., LATINO-40 Spanish Read News, Linguistic Data Consortium, Philadelphia, 1995.

Bimbot, F., Bonastre, J.-F., Fredouille, C., Gravier, G., Magrin-Chagnolleau, I., Meignier, S., Merlin, T., Ortega-García, J., Petrovska-Delacretaz, P. and Reynolds, D., 2004. A Tutorial on Text-Independent Speaker Verification. EURASIP Journal on Applied Signal Processing, 2004 (4), pp. 430-451.

Claes, T., Dologlou, I., Bosch, L. and Compernolle, D., 1998. A novel feature transformation for vocal tract length normalization in automatic speech recognition, IEEE Transaction on Speech and Audio Processing, 11 (6), 603–616, 1998.

Cui, X., and Alwan, A., 2006. Adaptation of children's speech with limited data based on formantlike peak alignment, Computer speech & language, 20(4), pp. 400-419, 2006.

Damper, R.-I. and Higgins, J.-E., 2003. Improving speaker identification in noise by subband processing and decision fusion. Pattern Recognition Letters, 24 (13), pp. 2167-2173.

Ding, G., Zhu, Y., Li, C. and Xu, B., 2002. Implementing vocal tract length normalization in the MLLR framework, In Proc. ICSLP 2002, pp. 1389–1392, 2002.

Eide and Gish, H., 1996. A parametric approach to vocal tract length normalization, in Proc. ICASSP 1996 , pp. 346–349, 1996.

Forsyth, M., 1995. Discriminating observation probability (dop) hmm for speaker verification. Speech Comm., vol. 17, pp. 117-129.

Furui, S., 2005. Recent progress in corpus-based spontaneous speech recognition. IEICE Transactions on Information and Systems, E88-D(3), pp. 366-375.

Giuliani, D., Gerosa, M. and Brugnara, F., 2006. Improved automatic speech recognition through speaker normalization, Computer Speech and Language, 20(1), pp. 107-123, 2006.

Heck, L.-P., Konig, Y., Kemal Sönmez, M. and Weintraub, M., 2000. Robustness to telephone handset distortion in speaker recognition by discriminative feature design. Speech Communication, 31(2-3), pp. 181-192.

Jelinek, F., 1997. Statistical Methods for Speech Recognition. Massachusetts Institute of Technology. Chapter 1-5. pp. 1-90.

Lamel, L. F., Rabiner, L. R., Rosenberg, A. E. and Wilpon, J. G., 1981. An improved endpoint detector for isolated word recognition. IEEE Trans. on Acoustics speech, and signal processing. Vol. ASSP- 29, pp. 777-785, August 1981.

Laurila, K., Vasilache, M. and Viikki, O., 1998. A combination of discriminative and maximum likelihood techniques for noise robust speech recognition. IEEE Conference on Acoustics, Speech and Signal Processing. pp. 12-15.

Lee, L., and Rose, R., 1998. "A frequency warping approach to speaker normalization, "IEEE Trans. Speech Audio Process., 6(1), pp. 49-60.

Leggetter, C. J. and Woodland, P. C., 1995. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models, Computer Speech and Language, 9, pp. 171–185, 1995.

Mak, M-.W., Tsang, C.-L., and Kung, S.-Y., 2004. Stochastic feature transformation with divergencebased out-of-handset rejection for robust speaker verification. EURASIP J. on Applied Signal Processing, vol. 2004, No. 4, pp. 452-465.

Oppenheim, A.-V., Willsky, A.-S. and Nawab, S.-H., 1997. Signals and Systems, Prentice Hall.

Openshaw, J.P., Sun, S.P. and Mason, J.S., 1993. A comparison of composite features under degraded speech in speaker recognition. Proceedings of ICASSP, Minneapolis, EE.UU., 2, pp. 371-374.

Panchapagesan, S. and Alwan, A., 2009. "Frequency warping for VTLN and speaker adaptation by linear transformation of standard MFCC," Computer Speech and Language, 23(1), pp. 42-46, 2009.

Picone, J., 1993. Signal Modeling Techniques in Speech Recognition. Proceedings of the IEEE, Vol. 81, No. 9, pp. 1215-1247.

Pitz, M., and Ney, H., 2005. Vocal Tract Normalization Equals Linear Transformation in Cepstral Space, IEEE Transactions on Speech and Audio Processing, 13(5-2), pp. 930-944, 2005.

Pitz, M., Molau, S., Schlüter, R., & Ney, H. 2001. Vocal Tract Normalization Equals Linear Transformation in Cepstral Space. In: Seventh european conference on speech communication and technology.

Rabiner, L. R., 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc. of IEEE, 77, 257--286.

Rabiner, L. R., Juang, B. H. and Lee, C. H., 1996. An Overview of Automatic Speech Recognition. Automatic Speech and Speaker Recognition: Advanced Topics, C. H. Lee, F. K. Soong and K. K. Paliwal editores, Kluwer Academic Publisher, 1996, pp. 1-30.

Reynolds, D.A., Zissman, M.A., Quatieri, T.F., O'Leary, G.C. and Carlson, B.A., 1995. The effects of telephone transmission degradations on speaker recognition performance. ICASSP'95.

Sanand, D.R., Schlüter, R., and Ney, H., 2010. Revisiting VTLN Using Linear Transformation on Conventional MFCC, In Proc. Interspeech 2010, pp. 538-541, 2010.

Savoji, M. H., 1989. A robust algorithm for accurate endpointing of speech signals. Speech Communication archive. Vol. 8, Issue 1, pp: 45 – 60. March 1989. Publisher Elsevier Science Publishers B. V. Amsterdam, The Netherlands.

Schwartz, R., Chow, Y. L., Kimbal, O., Roucos, S., Krasner, M., and Makhoul, J., 1985. Context-dependent modelling for acousticphonetic recognition of continuous speech. Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 1205-08, Trampa, FL, March 1985.

Skosan, M. and Mashao, D., 2006. Modified Segmental Histogram Equalization for robust speaker verification. Pattern Recognition Letters, 27 (5), pp. 479-486.

Thomas, S., Ganapathy, S., Hermansky, H., 2008. Recognition of Reverberant Speech Using Frequency Domain Linear Prediction. IEEE Signal Processing Letters, 15, pp. 681-684.

Tufekci, Z., 2007. Convolutional Bias Removal Based on Normalizing the Filterbank Spectral Magnitude. IEEE Signal Processing Letters, 14 (7), pp. 485-488.

Umesh, S., Zolnay, A. and Ney, H., 2005. Implementing frequency-warping and VTLN through linear transformation of conventional MFCC. In Proc. Interspeech 2005, pp. 269-272, 2005

Wakita, H., 1977. Normalization of Vowels by Vocal Tract Length and its Application to Vowel Identification. Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Vol. ASSP-25, No. 2, pp. 183–192, April 1977.

Wang, S., Cui, X., and Alwan, A., 2007. Speaker adaptation with limited data using regression-tree-based spectral peak alignment. IEEE Trans. Audio, Speech, Lang. Process., 15(8), pp. 2454-2464, 2007.

Wegmann, S., McAllaster, D., Orloff, J., and Peskin, B., 1996. Speaker normalization on conversational telephone speech. In Proc. ICASSP 1996 , pp. 339–341, 1996.

Wolfel, M., 2009. Enhanced Speech Features by Single-Channel Joint Compensation of Noise and Reverberation. IEEE Transactions on Audio, Speech, and Language Processing, 17 (2), pp. 312-323.