Identificación personal biométrica basada en bioseñales.

por:

Ing. Juan Carlos Atenco Vázquez

Tesis sometida como requisito parcial

para obtener el grado de:

MAESTRO EN CIENCIAS EN LA ESPECIALIDAD DE ELECTRÓNICA

en el:

INSTITUTO NACIONAL DE ASTROFÍSICA, OPTICA Y ELECTRÓNICA.

NOVIEMBRE 2018
Tonantzintla, Puebla.

Supervisada por:
Dr. Juan Manuel Ramírez Cortés
Investigador Titular, INAOE
Dr. René Octavio Aréchiga Martínez
Profesor Asociado, New Mexico Tech

RESUMEN

Actualmente hay una considerable cantidad de aplicaciones que requieren la verificación de la identidad del usuario tales como control de fronteras internacionales, aplicación de la ley y control de acceso a diferentes servicios, tanto privados como públicos. Métodos tradicionales de verificación de identidad incluyen tarjetas de identificación personal y contraseñas (las cuales el usuario necesita memorizar); sin embargo, estos métodos pueden ser fácilmente perdidos, olvidados o robados y por lo tanto comprometiendo la seguridad y, en la mayoría de los casos, la información personal del usuario.

En años recientes, los sistemas biométricos se han vuelto una solución viable al problema de la seguridad; operan bajo la premisa de que existen características fisiológicas y de comportamiento que son distintivas y permiten identificar a un individuo. Estas características son llamadas rasgos biométricos, son inherentes al cuerpo humano o al comportamiento de un individuo y por lo tanto los sistemas biométricos se apoyan en su unicidad para verificar la identidad de un usuario.

Este trabajo se enfoca en la creación de un sistema biométrico para verificar la identidad de los miembros de una población a través de las características espectrales de sus señales de voz, es decir, biometría de voz, también conocida como verificación de hablante. La razón para elegir este rasgo biométrico es porque la voz comprende las características tanto fisiológicas como de comportamiento de los hablantes, contiene mucha información sobre su identidad; y puede ser fácilmente grabada de un modo no intrusivo. Como tarea secundaria, se grabó una base de datos de voz en español basada en dígitos para entrenar y poner a prueba el sistema de verificación de hablante.

ABSTRACT

Currently there are a considerable amount of applications that require user authentication such as international border control, law enforcement and access control to different services, both private and public. Traditional authentication methods include ID cards and passwords (which the user needs to memorize); however, these methods can easily be lost, forgotten or stolen thereby compromising the security and, in most cases, the personal information of the user.

In recent years, biometric systems have become a viable solution to the security problem; it operates under the premise that there are distinctive physiological or behavioral characteristics that allow identifying an individual. These characteristics are called biometric traits, they are inherent to the human body or the behavior of an individual and therefore biometric systems rely on the uniqueness of these traits to authenticate the identity claim of a user.

This work focuses on the creation of a biometric system to verify the identity of the members a population through the spectral features of their signal voice, i.e. voice biometrics, also known as speaker verification. The reason to choose this biometric trait is that the voice comprises both physiological and behavioral features of speakers, contains lots of information about their identity; and can be easily recorded in a non-intrusive way. As a secondary task, a Spanish digit based voice database was recorded to train and test the speaker verification system.

AGRADECIMIENTOS

Agradezco de todo corazón a mi familia por su amor incondicional y constante apoyo durante esta etapa, en especial a mi abuelita Apolonia por siempre cuidar de mí.

Agradezco al Instituto Nacional de Astrofísica, Óptica y Electrónica y a sus investigadores por permitirme continuar con mi formación, al Consejo Nacional de Ciencia y Tecnología por el apoyo económico que me permitió concluir la maestría.

Agradezco sinceramente a mis asesores el Dr. Juan Manuel Ramírez Cortés y el Dr. René Aréchiga por brindarme la oportunidad de trabajar en este proyecto, por su tiempo, disposición y conocimientos para orientarme en esta etapa. También agradezco al M. C. Juan Carlos Moreno, al Dr. Rigoberto Salomón Fonseca, la Dra. María del Pilar Gómez Gil, el Ing. José Alfredo Jiménez Duarte y la Ing. Denisse Escarlette Mancilla Palestina, por sus valiosas contribuciones a este trabajo. Por último un agradecimiento especial al Dr. Carlos Zúñiga Islas por haberme permitido ingresar a INAOE durante mi periodo de prácticas profesionales lo cual me condujo a cursar la maestría.

Agradezco a mis amigos Sandra Báez y Ángel David Flores por su invaluable amistad y apoyo a lo largo de estos años.

Por último agradezco a Paloma por su amistad y cariño que me inspiraron a seguir luchando durante aquellos años tan difíciles.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ACRÓNIMOS

Bash	Born Again Shell.
DET	Detection Error Tradeoff.
EER	Equal Error Rate.
FAR	False Acceptance Rate.
FDC	Función de Distribución Acumulativa.
FRR	False Rejection Rate.
FPR	False Positive Rate.
HMM	Hidden Markov Model(s)
HTK	Hidden Markov Model Toolkit.
LLR	Log-Likelihood Ratio.
MAP	Maximmun A Posteriori.
MFCC	Mel Frequency Cepstral Coefficient(s).
MLE	Maximum Likelihood Estimation.
TPR	True Positive Rate.
UBM	Universal Background Model(s).

CAPÍTULO 1

BIOMETRÍA DE VOZ Y DEFINICIONES

1.1 OBJETIVOS DEL TRABAJO DE TESIS

1.1.1 OBJETIVO GENERAL

El presente trabajo de tesis tiene como objetivo realizar biometría de voz o reconocimiento de hablante con la tarea biométrica específica de verificación de hablante dependiente del texto en su variante de texto fijo. Los modelos clasificadores utilizados para construir el sistema de verificación son los Modelos Ocultos de Markov de izquierda a derecha con densidades continuas de observaciones, entrenados a nivel de palabra.

1.1.2 OBJETIVOS ESPECÍFICOS

Generar una base de datos en español orientada al desarrollo de experimentos de biometría en un ambiente sin ruido.

Realizar la extracción de características con base en los Coeficientes Cepstrales de Mel.

Entrenar los Modelos Ocultos de Markov para generar los modelos globales, y a partir de estos últimos generar los modelos de hablantes mediante una técnica de adaptación de parámetros.

Extender el sistema desarrollado con un esquema de comprobación de contraseña para añadir más seguridad al sistema.

Evaluar el sistema a través de figuras de mérito empleadas en biometría como la razón de verosimilitud logarítmica, la tasa de igual error (EER *Equal Error Rate*), las curvas ROC (*Receiver Operating Characteristics*) y DET (*Detection Error Tradeoff*).

1.2 INTRODUCCIÓN

La comunicación humana se establece de muchas maneras como lo son el lenguaje corporal, escrito, comunicación por imágenes y por voz. De estas formas de comunicarse la voz es una de las más básicas y considerada también como la más poderosa debido a la gran cantidad de información que se puede transmitir con ella.

En el día a día es común reconocer a una persona por su voz antes de iniciar una conversación telefónica; el aspecto de reconocimiento es muy importante ya que permite saber si se conoce a la otra persona y si tenemos confianza en ella para continuar la conversación, es decir, se necesita saber si es seguro conversar con la otra persona.

En niveles mayores de comunicación la seguridad es lo importante, porque la información que se transmite entre los individuos involucrados puede ser de carácter confidencial y el mal manejo de esta información puede tener consecuencias graves. Por lo tanto es necesario contar con métodos que garanticen que solo determinados individuos puedan transmitir y/o acceder a información delicada.

Para los propósitos de seguridad existen muchos métodos como contraseñas o claves PIN (Número de Identificación Personal) que son comunes para acceder a cuentas bancarias o de correo electrónico. Sin embargo, el principal problema de estos métodos y otros relacionados, es que son susceptibles a que el usuario olvide su clave de seguridad o en el peor de los casos que dicha clave sea robada, impidiendo el acceso a la información y comprometiendo la confidencialidad. Otro problema muy importante surge del hecho que estos métodos no pueden verificar ni comprobar que un usuario que ha ingresado correctamente una clave sea auténtico y no un impostor.

Para resolver la problemática descrita anteriormente existe la **Biometría,** que actualmente es una ciencia que va creciendo ante la necesidad de garantizar la seguridad de los usuarios para acceder a distintos servicios.

1.3 DEFINICIÓN DE BIOMETRÍA

En [26] se define **biometría** como la ciencia de **establecer la identidad** de un individuo basándose en los atributos fisiológicos o de comportamiento; establecer la identidad se interpreta como la identificación de un individuo de un conjunto de individuos o verificar que un individuo es quien dice ser.

La palabra biometría proviene del griego "**bios**" (vida) y "**metrikos**" o "**metría**" (medición), que se traduce como **medición de la vida** [27].

La biometría estudia la medición y análisis de aquellos atributos que se consideran únicos de un individuo y permiten diferenciar a uno de otro, a estos atributos se les conoce como **rasgos biométricos** [27].

1.4 RASGOS BIOMÉTRICOS Y SUS CARACTERÍSTICAS

Los rasgos biométricos se clasifican como **fisiológicos** o **de comportamiento**. Los primeros están asociados con la forma o las mediciones del cuerpo humano, mientras que los segundos, como su nombre lo indica, se asocian con el comportamiento de un individuo. La figura 1.1 muestra algunos ejemplos de rasgos biométricos tomando en cuenta su clasificación.

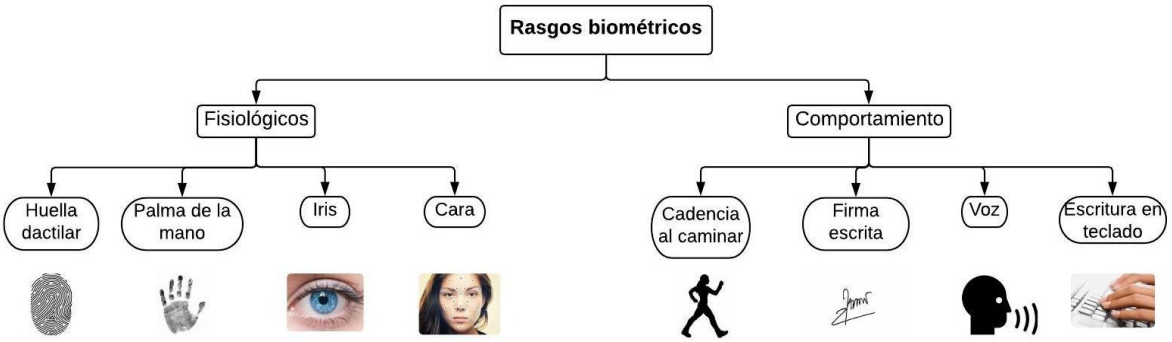


Fig. 1.1. Ejemplos de rasgos biométricos.

Las ventajas de usar rasgos biométricos para establecer la identidad de un individuo es que este mismo requiere estar físicamente presente al momento de llevar a cabo la identificación/verificación de su identidad y no necesita memorizar una contraseña o llevar consigo un objeto de identificación como una tarjeta.

Los rasgos biométricos tienen sus ventajas y desventajas. Dependiendo de los requerimientos de la aplicación se puede decidir entre un rasgo biométrico y otro, sin embargo, los rasgos biométricos tienen determinadas características que los hacen más o menos adecuados para una aplicación específica, estas características son [26]:

1. **Universalidad:** Todo individuo debe tener el mismo rasgo biométrico.

2. **Unicidad:** El mismo rasgo debe ser diferente entre los individuos.

3. **Permanencia:** El rasgo biométrico no debe cambiar durante un lapso de tiempo.

4. **Adquisición:** La facilidad con que se pueden medir, capturar y procesar los datos del rasgo biométrico.

5. **Desempeño:** La precisión del reconocimiento y los recursos para lograrla deben cumplir con las especificaciones de la aplicación.

6. **Aceptabilidad:** La población de usuarios debe tener la disponibilidad de presentar su rasgo biométrico al sistema.

7. **Circunvención:** La facilidad con que es posible imitar un rasgo de un individuo y por lo tanto engañar al sistema biométrico.

Cabe señalar que ningún rasgo biométrico cumple perfectamente cada característica. En la tabla 1.1 se incluyen tres niveles de satisfacción (Alto, Medio y Bajo) de las características de cada rasgo biométrico.

Tabla 1.1. Comparación de varios rasgos biométricos, A = alto, M = medio, B = bajo [13].

Rasgo biométrico	Universalidad	Unicidad	Permanencia	Adquisición	Desempeño	Aceptabilidad	Circunvención
Huella digital	M	A	A	M	A	M	M
Palma de la mano	M	A	A	M	A	M	M
Geometría de la mano	M	M	M	A	M	M	M
Iris	A	A	A	M	A	B	B
Retina	A	A	M	B	A	B	B
Cara	A	B	M	A	B	A	A
Oreja	M	M	A	M	M	A	M
Termograma	A	A	B	A	M	A	B
Voz	M	B	B	M	B	A	A
Firma escrita	B	B	B	A	B	A	A
Cadencia al caminar	M	B	B	A	B	A	M
Escribir en teclado	B	B	B	M	B	M	M
Olor	A	A	A	B	B	M	B
ADN	A	A	A	B	A	B	B
ECG	A	M	M	M	B	B	M

1.5 ESTRUCTURA DE UN SISTEMA BIOMÉTRICO

Independientemente de la elección de la característica para biometría el objetivo principal es crear un sistema completo que permita establecer la identidad de un individuo.

En [26] se define un **sistema biométrico** como un reconocedor de patrones que adquiere la información biométrica de un individuo, extrae un conjunto de datos a partir de la información biométrica, compara estos datos con los que ya están almacenados en una base de datos y ejecuta una acción basado en los resultados de la comparación. Un sistema biométrico genérico tiene los siguientes módulos:

1. **Módulo sensor:** sirve para capturar la información biométrica del individuo en forma de imágenes, audio, video o alguna otra señal. La calidad de la información capturada dependerá de la calidad del sensor.

2. **Módulo de extracción de vectores de características:** los datos biométricos son procesados para extraer **características discriminantes** que representen al rasgo capturado.

3. **Módulo de base de datos:** la base de datos es el módulo donde se almacena la información biométrica después de la etapa conocida como enrolamiento, esta información es procesada y con ella se crea un patrón o modelo. También se puede almacenar información biográfica de los individuos como nombre, edad, sexo, etc.

4. **Módulo de comparación:** los vectores de características extraídos se comparan con patrones o modelos que representen a los individuos registrados en el sistema biométrico, el resultado es una **calificación numérica** o *score*.

5. **Módulo de toma de decisión:** Los *scores* resultantes del módulo anterior son usados ya sea para validar una identidad o identificar a un individuo.

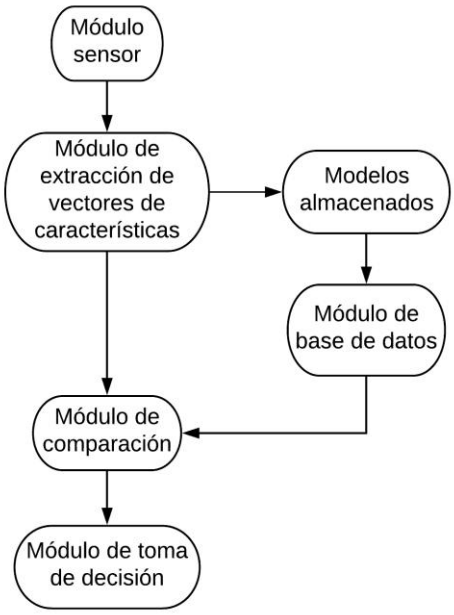


Fig. 1.2. Módulos de un sistema biométrico genérico [27].

En algunas fuentes como en [26] los módulos de comparación y de toma de decisión forman uno solo, en [27] (los módulos) están separados.

1.6 ENROLAMIENTO, VERIFICACIÓN E IDENTIFICACIÓN

El sistema biométrico tiene dos modos de operación: **enrolamiento** y **reconocimiento** [26]**.**

El **enrolamiento** es el proceso de extraer y almacenar en la base de datos la información de los rasgos biométricos de los individuos que componen la población para la cual se requiere un sistema biométrico. La información biométrica de un individuo una vez procesada se utiliza para generar un patrón o modelo mediante algún algoritmo y se almacena en el módulo de base de datos.

Normalmente la adquisición de datos requiere de varias sesiones de corta duración, esto le da más comodidad al usuario y por lo tanto está más dispuesto a cooperar.

En la literatura de biometría es común encontrar que se usa el término **reconocimiento** para referirse tanto a verificación como identificación de la identidad de un individuo. Dependiendo del contexto en el que se va a aplicar un sistema biométrico se especifica si se ejecutará la tarea de verificación o de identificación.

Verificación es un proceso de validar la identidad de un individuo comparando los datos biométricos extraídos con sus propios patrones o modelos almacenados en el sistema, es una comparación **uno a uno**.

Identificación es el mecanismo de comparar los datos biométricos de un individuo con los patrones o modelos de los otros individuos registrados en el sistema, es una comparación **uno a muchos**.

1.7 APLICACIONES DE BIOMETRÍA

Establecer la identidad de un individuo se ha vuelto crítico en un gran número de aplicaciones. La necesidad de técnicas de reconocimiento confiables se ha incrementado junto con la preocupación por mejorar los mecanismos de seguridad. En [26] y [27] estas aplicaciones se dividen en tres categorías:

1. Aplicaciones comerciales como comercio electrónico, banca en línea, acceso a internet, control de acceso físico, etc.

2. Aplicaciones gubernamentales como tarjetas de identificación oficial, seguridad social, etc.

3. Aplicaciones forenses como identificación de cuerpos, investigación criminal, determinación de paternidad, etc.

En [27] se encuentra una tabla que lista las ventajas y desventajas de varias técnicas de biometría.

1.8 BIOMETRÍA DE VOZ

Se debe entender por **biometría de voz** como la **identificación** y/o **verificación** de la identidad de un individuo mediante las **características de su voz** [13]. A este individuo se le refiere de forma concreta como **hablante**.

En esta modalidad de biometría los datos biométricos adquiridos durante el enrolamiento son señales de voz generadas por los usuarios al hablar. Las señales de voz contienen información fisiológica y de comportamiento del hablante. La figura 1.3 muestra el proceso de enrolamiento para biometría de voz.

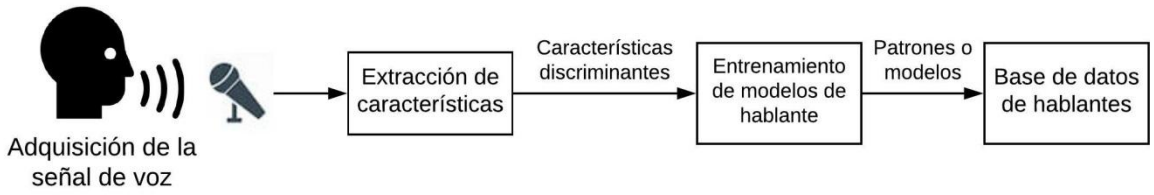


Fig. 1.3. Enrolamiento de hablantes para biometría de voz.

1.8.1 TERMINOLOGÍA Y DEFINICIONES

El término biometría de voz tiene muchas formas de ser referido en la literatura, el término más utilizado como sinónimo de biometría de voz es **reconocimiento de hablante**, el cual abarca la identificación y verificación**.**

Otros términos utilizados para referirse a biometría de voz es **reconocimiento de voz**, sin embargo, su significado está completamente enfocado al reconocimiento del contenido de la señal de voz (las palabras que se pronuncian). En la literatura en inglés se pueden encontrar otros términos para biometría de voz como ***speech biometrics***, ***voice identification biometrics*** o ***speaker identification***, que normalmente se consideran como sinónimos pero no son muy utilizados [3].

En este trabajo se referirá a biometría de voz como reconocimiento de hablante debido a que es el término que más se utiliza en la literatura y expresa mejor el objetivo de biometría de voz. Igualmente se usarán los términos **identificación de hablante** y **verificación de hablante** de forma explícita cuando sea necesario. El sistema biométrico de voz será llamado de manera general **sistema de reconocimiento de hablante**, en casos específicos se le llamará **sistema de identificación de hablante** o **sistema de verificación de hablante**.

Aquellos hablantes cuyos patrones o modelos están almacenados en el sistema biométrico se conocen como **hablantes genuinos** o **legítimos**, mientras que a estos mismos modelos que contienen su información biométrica se llaman **modelos de hablante**. Los hablantes que no están registrados en el sistema pero intentan validar su identidad haciéndose pasar como un hablante genuino se conocen como **impostores**.

A la muestra de voz que genera un hablante para registrarse en la base de datos o para utilizar el sistema de reconocimiento de hablante se le llamará **pronunciación** o **muestra de voz**.

Dado que la adquisición de datos biométricos se puede llevar a cabo con diferentes dispositivos de grabación de voz y en diferentes ambientes, se denomina **canal** al medio a través del cual se transmite la voz para ser grabada, por ejemplo, un canal puede ser un micrófono conectado una grabadora o la línea telefónica [3].

1.8.1.1 IDENTIFICACIÓN DE HABLANTE

Existen dos tipos de identificación: de **conjunto cerrado** (***closed-set***) y **de conjunto abierto** (***open-set***) [4].

En **identificación de conjunto cerrado**, la muestra de voz de un hablante de prueba se procesa y compara con todos los modelos de todos los hablantes genuinos, el identificador del modelo más parecido al hablante de prueba se entrega como resultado. Es importante señalar que en esta modalidad de identificación el sistema de identificación no decide si el hablante de prueba es quien dice ser.

En **identificación de conjunto abierto** además de entregar el identificador del modelo más parecido al hablante de prueba, también se hace una comparación entre ambos, si no hay un parecido satisfactorio se rechaza al hablante de prueba de lo contrario se acepta.

La figura 1.4 muestra la tarea de identificación de hablante.

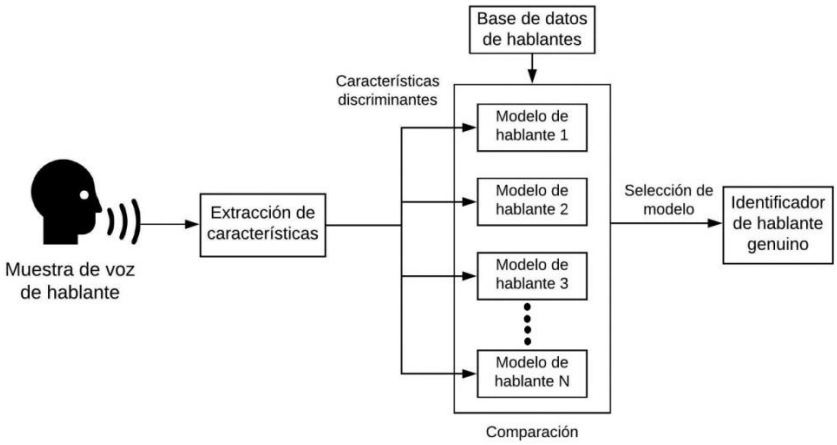


Fig. 1.4. Identificación de hablante.

1.8.1.2 VERIFICACIÓN DE HABLANTE

En **verificación de hablante** el hablante de prueba genera una pronunciación y se procesa, los datos se comparan solo con el modelo de hablante cuyo identificador corresponde al del hablante legítimo que el hablante de prueba dice ser. Si la comparación es satisfactoria, el individuo es verificado y aceptado, de lo contrario es rechazado.

La verificación de hablante es un caso especial de identificación de conjunto abierto con un solo hablante legítimo.

Existen diversas aproximaciones para asegurar que el hablante de prueba y un hablante legítimo sean lo más parecido posible. Dos aproximaciones muy importantes en la literatura introducen otro tipo de modelos. En capítulos posteriores se profundizará al respecto.

La figura 1.5 se ilustra la verificación de hablante.

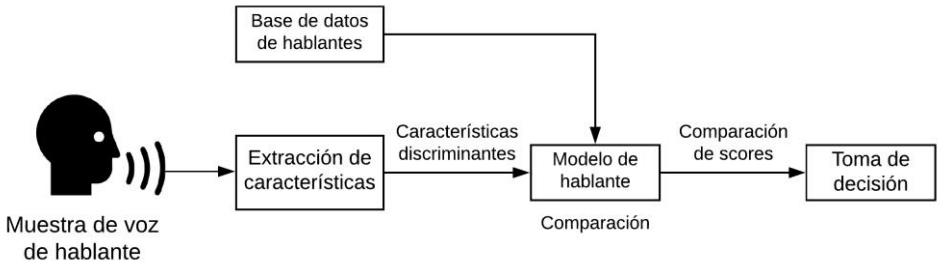


Fig. 1.5. Verificación de hablante.

1.9 MODALIDADES DE RECONOCIMIENTO DE HABLANTE

Comúnmente el reconocimiento de hablante es implementado usando dos modalidades vinculadas al lenguaje y al idioma: **reconocimiento de hablante dependiente del texto** e **independiente del texto** [3].

1.9.1 RECONOCIMIENTO DE HABLANTE INDEPENDIENTE DEL TEXTO

En un sistema de reconocimiento de hablante independiente del texto, el léxico de las muestras de enrolamiento de pruebas no está restringido [3].

Esta modalidad es apta tanto para verificación como para identificación, además requiere grandes cantidades de datos para ser implementado por lo que es necesario el uso de audio grabado de diversas fuentes. En aplicaciones que utilizan esta modalidad no se tiene control sobre el comportamiento del hablante.

1.9.2 RECONOCIMIENTO DE HABLANTE DEPENDIENTE DEL TEXTO

Esta modalidad solo puede ser aplicada a verificación [3], en ella hay un conjunto limitado de palabras (conocido como **léxico**) que es utilizado para llevar a cabo el reconocimiento de hablante, es decir, el reconocimiento está fuertemente ligado a las palabras pronunciadas por los hablantes genuinos.

El hecho de que el léxico esté limitado permite que las sesiones de enrolamiento sean muy cortas. Gracias a esto, esta modalidad es una buena opción para su implementación en aplicaciones comerciales.

Los sistemas de reconocimiento dependiente del texto se dividen en dos tipos:

- **Sistemas de texto fijo:** en estos sistemas el contenido léxico en el enrolamiento y de las muestras de voz para reconocimiento es siempre el mismo.
- **Sistemas de texto variable:** el contenido léxico de las muestras de voz para reconocimiento es diferente en cada intento de acceso al sistema.

1.10 INFORMACIÓN DE LA SEÑAL DE VOZ

La producción de voz, es un proceso extremadamente complejo cuyo resultado depende de muchas variables a diferentes niveles; entre estas variables se incluyen **factores sociolingüísticos** que influyen en las características biométricas de comportamiento y/o **factores fisiológicos** que hacen lo propio con las características físicas [26].

Estos dos conjuntos de factores afectan la creación de **patrones acústicos** que son únicos para cada individuo. Los patrones acústicos reflejan la anatomía (el tamaño y forma de la garganta y la boca) y los patrones aprendidos de comportamiento (tono de voz, forma de hablar) [13].

Las características fisiológicas del habla humana son invariantes para un individuo, pero la parte de comportamiento cambia con el tiempo debido a muchos factores como la edad o el estado emocional de un individuo [13].

1.10.1 NIVELES DE INFORMACIÓN

Los seres humanos reconocen a un hablante por una combinación de diferentes niveles de información, los cuales son diferentes para cada hablante [26].

- En el nivel más alto está el **idioléctico**, que es la forma particular que tiene un individuo de hablar un idioma. Esta forma está influenciada por el nivel educativo, sociológico y la ubicación geográfica.

- En el siguiente nivel se encuentran las características **fonotácticas**, que describen el uso de un hablante de las unidades lingüísticas conocidas como fonemas. Cada hablante tiene patrones de uso que lo distinguen.

- En un tercer nivel están las características **prosódicas**, que son la combinación de energía instantánea, entonación, ritmo al hablar y tono emocional.

- En el nivel más bajo, se encuentran las características **espectrales** de la señal de voz. La información espectral pretende extraer las características del tracto vocal y la dinámica de articulación de palabras. Existen dos tipos de información de bajo nivel: en información **estática** la señal de voz es segmentada en tiempo y se analiza cada segmento, la información **dinámica** se relaciona con la forma en que la señal de voz cambia a través del tiempo y toma en cuenta el proceso en el cual un individuo pasa de una articulación de una palabra a otra.

.

CAPÍTULO 2

EXTRACCIÓN DE CARACTERÍSTICAS DE LA SEÑAL DE VOZ

2.1 PROCESO DE PRODUCCIÓN DE LA VOZ

Los órganos que participan en la producción de la voz son parte del aparato respiratorio y del sistema digestivo, también se ven involucrados procesos del cerebro que nos permiten controlar dichos órganos para generar sonidos de manera articulada, esos sonidos articulados son la **voz**.

En [3] y [4] se describen exhaustivamente los órganos involucrados para generar la voz, la forma en que interactúan entre sí, inclusive los procesos cerebrales que permiten la que los sonidos formen palabras. En este apartado solo se describirá brevemente el proceso para producir la voz.

El **aparato fonador** es el conjunto de órganos de nuestro cuerpo encargados de producir y amplificar los sonidos del habla. Los órganos se dividen en **órganos de fonación** encargados de la producción de sonido: la cavidad laríngea o glótica, las cuerdas vocales y los resonadores que consisten en las cavidades nasal, oral y la faringe. Por otra parte están los **órganos de articulación**, que modifican el sonido producido por efecto de los movimientos que hacen, los principales órganos son: el paladar, la lengua, la úvula, los dientes, los labios y la glotis. Los pulmones, los bronquios y la tráquea también son órganos muy importantes, ya que suministran y conducen el aire a los otros órganos durante la exhalación. La figura 2.1 muestra la ubicación de los órganos antes mencionados en el cuerpo humano.

La generación de la voz humana es un proceso que combina el flujo de aire que viene desde los pulmones. Este flujo de aire pasa a través de la tráquea y después a la cavidad laríngea, en esta cavidad pasa por las cuerdas vocales para generar una oscilación y por lo tanto un sonido; estas últimas se tensan para producir sonidos agudos y se relajan cuando se producen sonidos graves. Es en esta parte donde se ajustan el tono y el volumen de la voz.

Posteriormente los órganos articulatorios alteran el sonido producido por los órganos de fonación moviéndose en distintas posiciones para producir sonidos más definidos que componen las palabras que pronunciamos al hablar.

Si el aire que sale de la boca no fue obstaculizado o modificado por los órganos articulatorios el sonido es una **vocal**; en caso contrario de que si hayan intervenido el sonido producido es una **consonante**.

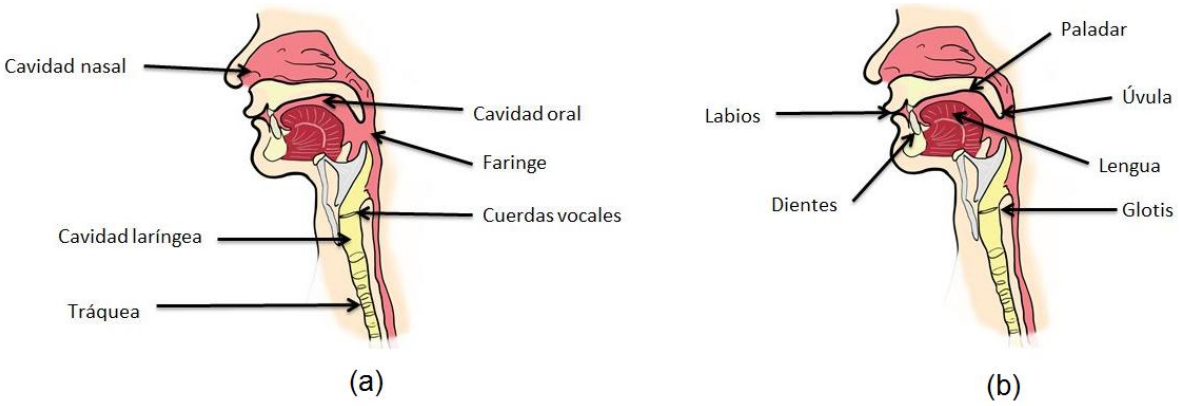


Fig. 2.1. Órganos del aparato fonador. (a) Órganos de fonación. (b) Órganos de articulación. [3]

2.2 LA SEÑAL DE VOZ

En [3] se define **señal** como **la medición de un fenómeno físico observado**. Matemáticamente hablando se puede definir como una **función**, la cual tiene un *dominio* donde está definida y su correspondiente rango.

En una señal puede haber dos variables que describen a un fenómeno físico determinado, frecuentemente el **tiempo** es una de las variables, aunque también puede ser el espacio, mientras que otra variable puede ser el fenómeno físico que se está midiendo en sus unidades correspondientes. Por lo tanto la señal es un mapeo de una variable como el tiempo dentro de otra variable como puede ser voltaje, corriente, etc.

La señal de voz básicamente es la medición de la amplitud de las ondas de la voz con respecto al paso del tiempo. Está medición de amplitud no está relacionada a una sola frecuencia específica, sino que dentro de la señal hay una combinación de muchas frecuencias, en el habla humana se considera que las frecuencias van de aproximadamente 250 a 3000 Hz [4], es por esto que se considera que la señal de voz contiene una gran cantidad de información. La figura 2.2 muestra gráficamente una señal de voz.

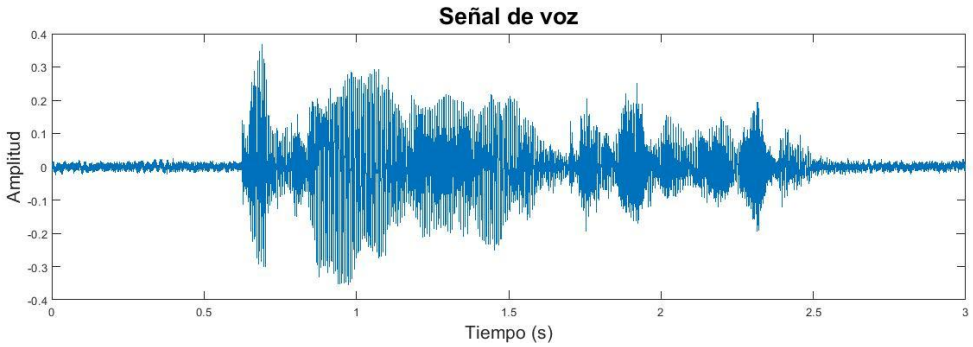


Fig. 2.2. Gráfica de una señal de voz.

2.2.1 CARACTERÍSTICAS DE LA SEÑAL DE VOZ

Debido a que la señal de voz es generada por el cuerpo humano tiende a cambiar mucho en el tiempo, por lo tanto se dice que es **dinámica y no estacionaria** (los parámetros estadísticos cambian con el paso del tiempo). Si la señal de voz es no estacionaria entonces también es **no periódica** por definición [3].

2.3 EXTRACCIÓN DE VECTORES CARACTERÍSTICAS

Las características espectrales son las que se extraerán de la señal de voz. El resultado esperado es un conjunto de vectores de números que representan a la señal de voz.

2.3.1 MÉTODOS DE EXTRACCIÓN DE VECTORES DE CARACTERÍSTICAS

La señal de voz por sí misma no es útil para operar sobre ella debido a sus cualidades, mencionadas en el apartado 2.2.1. Para resolver este problema se han creado diversos métodos matemáticos que tienen como objetivo **extraer características discriminantes** de la señal de voz, dichas características dependen del enfoque particular del método utilizado, el resultado de realizar esta operación es un vector o conjunto de vectores que representan a la señal de voz, a estos vectores se les llama **vectores de características** [3].

Algunos métodos matemáticos de extracción de características se pueden consultar en [11], [14], [19] y [31], en la tabla 2.1 se listan algunos de estos métodos y su enfoque.

Tabla 2.1. Listado de algunos métodos de extracción de características para la señal de voz y su enfoque particular.

Método de extracción de características	Enfoque
LDB (Local Discriminant Bases)	Identificar subespacios tiempo-frecuencia discriminatorios de la señal de voz.
MFCC (Mel Frequency Cepstral Coefficients)	Obtener vectores de coeficientes que representan la tendencia logarítmica de la audición humana.
PCA (Principal Component Analysis)	Reducir las dimensiones de un vector(es) de características existente, resaltando solo los componentes más importantes de ese vector(es).
LPCC (Linear Prediction Coding Coefficients)	Un método de autocorrelación para imitar el habla humana.
PLP (Perceptual Linear Prediction)	Derivar un espectro con características similares a las de la audición humana.
RASTA PLP	Aplica un filtro pasa banda a las sub bandas de frecuencia de PLP para eliminar variaciones de ruido en tiempos cortos.
i-vector	Compensar la variabilidad de la voz en información utilizada para entrenamiento y prueba de modelos de hablante.
Bottleneck	Filtra la información de una señal de voz y toma las partes más importantes.

Como se puede observar hay una gran variedad de métodos para extraer características, que se concentran ya sea en alguna cualidad particular de la señal de voz o en la forma en que el ser humano la percibe. También es posible hacer combinaciones entre métodos, aunque su efectividad dependerá, entre otras cosas, del modelo clasificador utilizado.

Seleccionar un método de extracción de características (o una combinación de métodos) es una las partes más importantes en biometría de voz, ya que la decisión impactará significativamente en los resultados. En este trabajo se eligió trabajar con los coeficientes ***MFCC (Mel Frequency Cepstral Coefficients*** o **Coeficientes Cepstrales en Frecuencia de Mel** en español***)***.

2.4 REVISIÓN DE LITERATURA SOBRE LOS COEFICIENTES CEPSTRALES EN FRECUENCIA DE MEL (MFCC)

La decisión de elegir los MFCC se debe a que han demostrado gran eficacia en este tipo de aplicaciones. En [2] y [33] se combinan los MFCC con otros vectores de características para entrenar redes de creencia profunda y máquinas de soporte vectorial respectivamente, ambos para reconocimiento de hablante; mientras que en [22] se utilizan para entrenar modelos de mezclas gaussianas para reconocer hablantes de lenguaje Hindi.

Por otra parte en [19] se muestra una comparación de resultados al utilizar los MFCC junto con otros vectores de características para entrenar y probar diferentes tipos de modelos acústicos para reconocimiento de hablante; la comparación muestra que los MFCC, al usarlos en diferentes tipos de pruebas de reconocimiento, tienen un buen desempeño al reportar el menor porcentaje de error comparado con los porcentajes de error resultantes de concatenar los MFCC con otros vectores de características; y aunque también se reportan pruebas en las que los MFCC tienen mayor porcentaje de error que los vectores de características concatenados, ese porcentaje de error sigue siendo bajo.

En [8] se comparan los porcentajes de error y de exactitud para una tarea de reconocimiento de hablante dependiente del texto utilizando diferente número de coeficientes MFCC, se observa que para más de 14 coeficientes la exactitud de reconocimiento es de mayor de 90%, en cuanto al porcentaje de error para más de 14 coeficientes el valor es menor de 1%, lo que demuestra que los MFCC son una excelente opción para la tarea de reconocimiento de hablante.

También vale la pena revisar el desempeño de los MFCC para otras tareas de reconocimiento, por ejemplo en [14] y [35] se utilizan para reconocimiento de voz, que es un primer paso para el reconocimiento de hablante. En [14] se demuestra que los MFCC concatenados con su primera y segunda derivada tienen un porcentaje de 98% de reconocimiento de voz, al compararlos con los otros vectores de características y combinaciones de estos, se aprecia que todos tienen un porcentaje de reconocimiento similar. Por otro lado en [35] se estudian los porcentajes de reconocimiento de voz para diferentes configuraciones de coeficientes MFCC en combinación con diferentes configuraciones de parámetros de **Modelos Ocultos de Markov** (**HMM** por sus siglas en inglés), se destaca el hecho de que los MFCC con sus derivadas tienen el mejor porcentaje de reconocimiento de voz para cada configuración de parámetros de Modelos

Ocultos de Markov comparándolos con MFCC y una sola derivada, con MFCC con coeficiente de energía y solo los coeficientes MFCC.

Con estos antecedentes se respalda la decisión de elegir los MFCC como los vectores de características de voz para este trabajo, ya que son flexibles para realizar distintas tareas de reconocimiento en el área de voz, también se pueden combinar con otros vectores de características y tener altos porcentajes de reconocimiento, además de poder utilizarlos en conjunción con diferentes modelos acústicos para clasificación.

2.5 LOS COEFICIENTES CEPSTRALES EN FRECUENCIA DE MEL (MFCC)

Para comenzar a definir con detalle los MFCC primero se definirá el concepto de ***cepstrum*** [4].

Cepstrum es una palabra resultante de invertir las primeras cuatro letras de *spectrum* (espectro en inglés), la definición más común es:

$$c\ n\ = \frac{1}{2\pi}\ \int_{-\pi}^{\pi} \log X\ e^{i\omega}\quad e^{i\omega n} d\omega. \tag{1}$$

La ecuación (1) indica que el *cepstrum* es la inversa de la transformada discreta de Fourier del logaritmo base 10 de la magnitud de la transformada de Fourier de tiempo discreto $X(e^{i\omega})$ de una señal.

Mientras que el espectro de frecuencia, obtenido mediante la transformada de Fourier de tiempo discreto, describe la amplitud y la fase de cada banda de frecuencia, el *cepstrum***,** obtenido de aplicar otra transformada al espectro de frecuencia, caracteriza variaciones entre las bandas de frecuencia, esta característica del *cepstrum* es útil para describir la señal de voz.

Sin embargo, es importante mencionar que si bien la ecuación (1) es una definición común para calcular el *cepstrum*, en procesamiento de voz es común que se obtenga aplicando la **transformada discreta de coseno** en lugar de la inversa de la transformada discreta de Fourier, la motivación para esto se explicará en apartados posteriores.

Existen varios métodos para extraer características de una señal de voz que están basados en el *cepstrum* y son principalmente usados en señales de voz, los MFCC son los más populares para tareas de reconocimiento de voz y de hablante. Una ventaja de estos vectores de características es que son más compactos, discriminantes y prácticamente sin correlación entre ellos [18].

Los MFCC están pensados para tomar en cuenta la **forma en que la audición humana percibe el sonido**. Estudios fisiológicos muestran que la **banda de frecuencias crítica** de la audición humana se extiende hasta 1 KHz, las frecuencias en esta banda son percibidas de forma lineal, mientras que por encima de esta frecuencia la percepción se vuelve logarítmica [5] [12].

Por lo anterior se dice que la información más importante del sonido se encuentra en las bajas frecuencias, comúnmente en la práctica se calculan 13 o más coeficientes,

pero solo se conservan los primeros 13 porque contienen toda la información de las frecuencias bajas.

Para incluir la cualidad logarítmica del sonido dentro de los MFCC se implementa un concepto conocido como banco de filtros en la llamada escala de Mel, estos conceptos se explicarán más adelante.

En los siguientes apartados se describe el proceso para extraer los MFCC de una señal de voz.

2.5.1 PREPROCESAMIENTO DE LA SEÑAL DE VOZ

Para comenzar el proceso de extraer los MFCC la señal de voz ya debió ser almacenada en un formato sobre el que se pueda operar, lo más común es que se grabe mediante un micrófono, muestreando la señal con una **frecuencia de muestreo** acorde con el **teorema de Nyquist-Shannon.** Con este primer procedimiento la señal de voz está discretizada, por lo cual está representada por una cantidad finita de valores, a estos valores se les conoce como **muestras**; la cantidad de muestras que representan la señal de voz depende de la frecuencia de muestreo y del tiempo de duración de la señal [3].

Es recomendable aplicar un filtro pasa altas a la señal de voz discretizada para compensar por la información contenida en las altas frecuencias de la señal de voz que normalmente se atenúan durante la producción de la voz por el cuerpo humano [36]. El filtro pasa altas puede ser de primer orden, si *x[n]* es una señal de voz el filtro tiene la siguiente ecuación:

$$y[n] = x[n] - \alpha x[n-1], \tag{2}$$

donde α es el coeficiente del filtro, cuyos valores típicos son 0.95 o 0.97.

En el apartado 2.2.1 se mencionó que una cualidad de la señal de voz es que es no estacionaria, sin embargo, se encontró que en segmentos de tiempo pequeños se puede considerar estacionaria, estos tiempos cortos se encuentran en el rango de 10 a 30 milisegundos [5], en la práctica 25 milisegundos es el tiempo más utilizado.

Tomando en cuenta la propiedad anteriormente descrita, el primer paso para calcular los MFCC es segmentar la señal de voz, si *x[n]* es una señal de voz discretizada se divide en *P* segmentos, cada segmento tiene *N* muestras (estas *N* muestras deben ser el equivalente a un tiempo de entre 10 y 30 milisegundos), hay un traslape entre segmentos el cual comúnmente es de 50%, es decir, la primer muestra del segmento *p* se encuentra en la muestra $\frac{N}{2}$ del segmento *p-1*, sin embargo, el traslape se puede realizar en cualquier muestra de un segmento, la figura 2.3 ejemplifica la segmentación de una señal de voz con traslape de 50%. Si el total de muestras que conforma una determinada señal de voz no es suficiente para completar los *P* segmentos, las muestras faltantes se completarán con ceros.

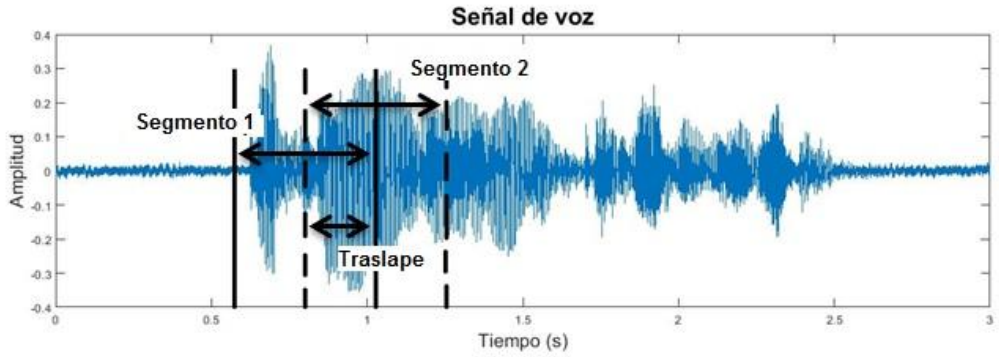


Fig. 2.3. Ejemplo de segmentación y traslape entre segmentos de una señal de voz.

La señal segmentada se representará como:

$$\chi = \{x_1[n], x_2[n], x_3[n], \ldots, x_p[n], \ldots, x_P[n]\}, \quad (3)$$

donde $x_p[n]$ es un segmento de la señal y como se mencionó anteriormente N es el número de muestras de un segmento p, con $p = 1, 2, 3, \ldots, P$; y n es una muestra de ese mismo segmento con valores $n = 0, 1, 2, \ldots, N-1$.

El segundo paso es realizar un ventaneo, se multiplican las muestras de un segmento p por una **ventana Hamming** cuya ecuación es:

$$w\ n\ = 0.54 - 0.46\cos(\frac{n\pi}{N}). \quad (4)$$

Se observa que la ventana Hamming tiene el mismo número de muestras N que los segmentos en que se dividió la señal $x[n]$.

El ventaneo minimiza las discontinuidades de la señal reduciendo la magnitud de las primeras y últimas muestras del segmento a cero [35]. La figura 2.4 muestra un ejemplo de ventana Hamming.

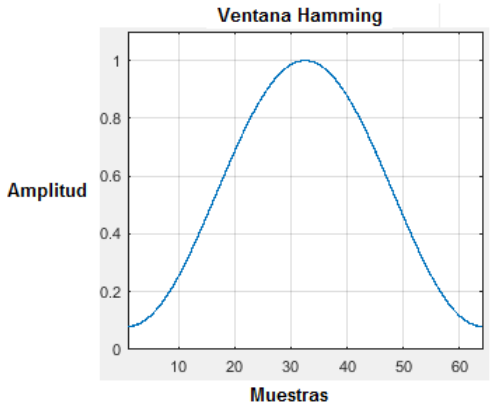


Fig. 2.4. Ventana Hamming.

2.5.2 TRANSFORMADA DISCRETA DE FOURIER Y MAGNITUD DEL ESPECTRO DE FRECUENCIAS

El tercer paso es aplicar la **transformada discreta de Fourier** a todos los segmentos de señal al cual se le ha aplicado el ventaneo, con esto se obtiene el espectro de frecuencia de cada ventana de la señal de voz, de acuerdo a la siguiente ecuación:

$$X_p\ k\ = \sum_{n=0}^{N-1} x_p[n]\, w[n] e^{-\frac{i2\pi kn}{N}}, \tag{5}$$

donde $x_p[n]\ w[n]$ es la multiplicación de los valores de un segmento de señal por los valores de la ventana Hamming; $X_p(k)$ es la transformada discreta de Fourier, k es una muestra de la transformada discreta de Fourier de longitud K, con $k = 0, 1, 2, \ldots, K-1$.

Cuando se calcula la transformada discreta de Fourier a todos los segmentos ventaneados se tiene una matriz $X = [X_1\ k\ , X_2\ k\ , X_p\ k\ , \ldots , X_P\ k\ ]$, posteriormente se obtiene la magnitud de todos los componentes de X, es decir el **espectro de magnitud** $|X|$.

2.5.3 EL BANCO DE FILTROS EN FRECUENCIA DE MEL

Como se mencionó en el apartado 2.5, los MFCC están modelados tomando en cuenta la percepción logarítmica del sonido por el ser humano, para este propósito se lleva a cabo un análisis en frecuencia el cual convierte la frecuencia en Hz a la **escala de Mel** y viceversa, acuerdo con las siguientes ecuaciones:

$$Mel\ f\ = 2595\ log_{10}(1 + \frac{f}{700}), \tag{6}$$

$$f\ m\ = 700\ (10^{\frac{m}{2595}} - 1), \tag{7}$$

donde f es la frecuencia en Hz y m es la frecuencia en escala de Mel.

El espectro de magnitud $|X|$ se segmenta en bandas de frecuencia críticas con un **banco de filtros** espaciados linealmente a frecuencias menores a 1 KHz y logarítmicamente a frecuencias mayores. Este banco de filtros tiene una respuesta en frecuencia pasa banda de forma triangular, están traslapados al 50%; tanto el espaciado entre filtros como el ancho de banda de cada uno están determinados por un intervalo de frecuencia en escala de Mel, para cada filtro corresponde un componente en escala de Mel [31]. La figura 2.5 muestra un ejemplo de banco de filtros en escala de Mel.

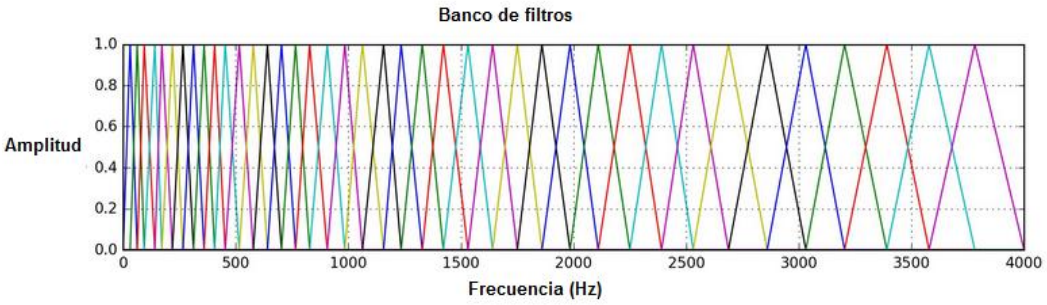


Fig. 2.5. Banco de filtros en escala de Mel.

El banco de filtros tiene *F* filtros, con una frecuencia inferior f_{min} y una frecuencia superior f_{max}, para una señal de voz se recomienda que f_{min} sea superior a 100 Hz para no tener interferencia por parte de la frecuencia de línea eléctrica. Por otro lado f_{max} se recomienda que sea menor a la frecuencia de Nyquist, a frecuencias mayores a 6800 Hz ya no hay mucha información en la señal de voz [18].

En [18] se detalla el cálculo de las frecuencias inferior, central y superior de cada filtro, así como la forma de espaciarlos linealmente a frecuencias inferiores a 1 KHz y logarítmicamente a frecuencias superiores.

2.5.4 CÁLCULO DEL CEPSTRUM DE MEL

Una vez calculado el banco de filtros *M(m,k),* donde *m* es un filtro del banco; y con *m = 1, 2, 3,…, F;* se procede a multiplicar el banco de filtros por el espectro de magnitud y calcular el logaritmo base diez del producto, como se observa en la siguiente ecuación:

$$L_p\ m,k\ = log_{10}\ \ \sum_{k=0}^{K-1} M\ m,k\ *\ X_p(k)\ \ . \tag{8}$$

El resultado de la multiplicación del banco de filtros con el *espectro de magnitud* es conocido como el **espectro de Mel** [18].

Como se mencionó en el apartado 2.5 para calcular el *cepstrum* es necesario aplicar la inversa de la transformada de Fourier una vez que se calculó el logaritmo de base 10 a la transformada de Fourier de la señal de voz. Pero en el caso de los MFCC la segunda transformada que se aplica es la **transformada discreta de coseno** por las siguientes razones:

- Convierte el espectro en escala de Mel al dominio del tiempo.
- Por sus características es ideal para comprimir la información, por lo que habrá menos coeficientes resultantes en comparación a usar la inversa de la transformada discreta de Fourier, de los cuales los primeros son los que contendrán una mayor cantidad de información de la señal de voz.

- Al aplicar la *transformada discreta de coseno* los coeficientes resultantes solo tienen valores reales.

La siguiente ecuación describe la aplicación de la transformada discreta de coseno al espectro logarítmico de Mel:

$$\Phi_p^r\ x[n] \ = \ \sum_{m=1}^{F} L_p(m,k)\ cos\ \frac{r(2m-1)\pi}{2F}\ , \tag{9}$$

donde *r* es un número de coeficiente, con *r = 1, 2,…, F;* y $\Phi_p^r\ x[n]$ es el coeficiente *r* del segmento *p* de la señal de voz *x[n],* cabe mencionar que el número de filtros del banco es igual al número de coeficientes por segmento.

Los MFCC resultantes serán una matriz *FxP***,** lo que indica que habrá *F* coeficientes por cada segmento en que se haya dividido la señal de voz *x[n].* Está matriz de MFCC se representa como:

$$\Phi[\chi] = \ \Phi_1, \Phi_2, \dots, \Phi_p, \dots, \Phi_P\ , \tag{10}$$

donde Φ_p es un vector de *F* coeficientes de Mel correspondiente al segmento $x_p[n]$.

El procedimiento completo para hacer el cálculo de los MFCC se puede resumir como se muestra en la figura 2.6.

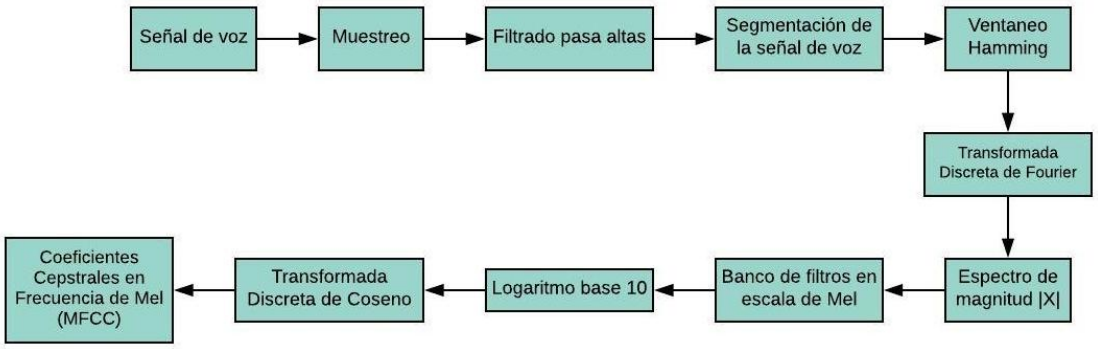


Fig. 2.6. Procedimiento para extraer los MFCC de la señal de voz.

2.6 COEFICIENTES DIFERENCIALES: DELTA Y ACELERACIÓN

Los MFCC por si mismos son una excelente representación de la señal de voz, sin embargo, es posible mejorar el reconocimiento agregando **coeficientes diferenciales**, estos capturan las características dinámicas de la voz [3].

Comúnmente solo se utilizan la primera y segunda derivadas aplicadas a los coeficientes estáticos de los MFCC, la primera derivada es conocida como **Delta** mientras que la segunda derivada se conoce como **Delta-Delta** o de **Aceleración**. El número de coeficientes Delta y Delta-Delta que se utilizan puede ser igual al de los MFCC, en [3] se menciona que es posible utilizar menos coeficientes Delta y Delta-Delta comparados al número de MFCC.

Para calcular los coeficientes de primera derivada se utiliza la siguiente ecuación:

$$d_p = \frac{\sum_{\theta=1}^{\Theta} \theta (c_{p+\theta} - c_{p-\theta})}{2 \sum_{\theta=1}^{\Theta} \theta^2}, \quad (11)$$

donde d_p es un coeficiente *delta* de un segmento p dados los coeficientes estáticos adyacentes $c_{p+\theta}$ y $c_{p-\theta}$, Θ son la cantidad de segmentos adyacentes a p y θ es un segmento adyacente a p. Para los casos en que los segmentos de referencia p son el primero o el último se toman las siguientes consideraciones:

- Si $p = 1$ (el segmento de referencia es el primero) entonces $p - 1 = P$ (el segmento adyacente izquierdo será el último segmento de la señal de voz), por lo tanto se tiene que $p - 2 = P - 1$ y así sucesivamente.

- Si $p = P$ (el segmento de referencia es el último) entonces $p + 1 = 1$ (el segmento adyacente derecho será el primer segmento de la señal de voz), por lo tanto se tiene que $p + 2 = 2$ y así sucesivamente.

Para obtener los coeficientes Delta-Delta se aplica la ecuación 11 a los coeficientes Delta.

CAPÍTULO 3

MODELOS OCULTOS DE MARKOV Y MODELOS DE HABLANTES

3.1 CLASIFICADORES PARA RECONOCIMIENTO DE HABLANTE

En reconocimiento de hablante, reconocimiento de voz y otras áreas afines frecuentemente se buscan secuencias de objetos, por ejemplo una secuencia de vectores de características. También es necesario modelar la duración de la secuencia de salida así como el contenido de la misma [3].

Generalmente se asume que la señal de voz está codificada como una secuencia de uno o varios símbolos. Entonces para reconocer ya sea a un hablante o el contenido de un mensaje de voz, se debe reconocer esa secuencia de símbolos convirtiéndola primero en un vector de características. Posteriormente un modelo clasificador hace un mapeo entre los vectores de características de una o varias señales de voz y la secuencia de símbolos para entregar un resultado que mida la comparación de acuerdo con un criterio preestablecido. En el caso de verificación de hablante se toma una decisión de rechazar o validar la identidad de una persona y en reconocimiento de voz se entregan secuencias de palabras que muy probablemente representan el contenido de un mensaje de voz [36].

La elección de un **clasificador** depende de las condiciones y limitaciones de la aplicación, así como del tipo de vector de características que se utilizan para entrenarlo y probarlo.

En el área de reconocimiento de hablante el objetivo principal es utilizar un clasificador que modele a un hablante, lo que significa que este clasificador será entrenado utilizando los vectores de características obtenidos de las señales de voz generadas por un hablante. De esta forma el clasificador actuará como una representación del hablante mismo y tendrá las mismas características acústicas de su voz. Este clasificador entrenado se conoce como **modelo de hablante**.

En [4] se dividen los métodos para modelar hablantes en **paramétricos** y **no paramétricos**.

Los métodos **no paramétricos** hacen pocas suposiciones estructurales sobre la información y necesitan datos de entrenamiento que sean comparables con los datos de prueba, ejemplos de estos métodos son el uso de **patrones de hablante** junto con el algoritmo de **Alineamiento Temporal Dinámico***;* o modelado de **Vecino más cercano**. Por otro lado los métodos **paramétricos** toman en cuenta las limitaciones relacionadas con la información, ejemplos de estos métodos son los **Modelos Ocultos de Markov, Máquinas de Soporte Vectorial** o **Mezclas Gausianas Mixtas***.*

En el siguiente apartado se hace una revisión breve de la literatura sobre clasificadores utilizados para modelar hablantes.

3.2 REVISIÓN DE LITERATURA SOBRE CLASIFICADORES PARA RECONOCIMIENTO DE HABLANTE

En [30], [29], [16] y [8] se encuentran revisiones de literatura sobre clasificadores que se utilizan en **verificación de hablante dependiente del texto**. Destacan las menciones de **Redes Neurales Profundas** tanto como clasificador como extractor de características. Otro clasificador destacado es el llamado ***HiLam*** (modelo acústico multicapa jerárquico) que es una combinación de **Mezclas Gausianas Mixtas** y **Modelos Ocultos de Markov**. También se mencionan **Redes Neurales Artificiales**, **Máquinas de Soporte Vectorial**, **Alineamiento Temporal Dinámico**, **Cuantización vectorial**, entre otros.

Otro punto muy importante a resaltar es que en la revisión de literatura que hacen estos cuatro artículos, se mencionan constantemente los Modelos Ocultos de Markov y las Mezclas Gausianas Mixtas, ambos modelos clasificadores se utilizan con muchos tipos de vectores de características y sus implementaciones se hacen con muchas variantes en el campo léxico, además es frecuente que sean combinados con otros *clasificadores* como en el caso del ya mencionado *HiLam*.

Para ejemplificar el amplio uso de los Modelos Ocultos de Markov y las Mezclas Gausianas Mixtas dentro del estado del arte del reconocimiento de hablante se pueden consultar los trabajos [30], [29] y [19]. En [29] y [30] se utilizan ambos clasificadores para una tarea de verificación de hablante dependiente del texto, los resultados reportados muestran bajos porcentajes de error para distintos tipos de pruebas que involucran impostores tratando de vulnerar el sistema.

En el apartado 2.4 se mencionó que en [19] se hace una comparación sobre los resultados de reconocimiento de hablante para distintos vectores de características, también se hace una comparación de sistemas para verificación de hablante que utilizan Modelos Ocultos de Markov, Mezclas Gausianas Mixtas y una combinación de ambos clasificadores, se realizan distintos tipos de pruebas con impostores utilizando distintos léxicos de una misma base de datos; los resultados también muestran bajos porcentajes de error.

Para elegir un clasificador que modele a un hablante también se debe considerar su flexibilidad para ser utilizado en otras tareas de reconocimiento donde se involucra la señal de voz. En este rubro los Modelos Ocultos de Markov son muy utilizados desde hace mucho tiempo como en [24]. En el caso de reconocimiento de voz se reporta en [15] un sistema de reconocimiento de voz que utiliza Modelos Ocultos de Markov para lenguaje arábigo; en [35] también se utilizan para reconocimiento de voz; en [14] se utilizan estos modelos en combinación con diferentes vectores de características para reconocimiento de voz.

En tareas de reconocimiento dependientes del texto, saber de antemano el texto hablado en las muestras de voz de enrolamiento y prueba; puede ser combinado con información temporal que nos aportan los HMM [4], [25] lo cual supera el desempeño de modelos como Mezclas Gaussianas Mixtas.

Tomando en cuenta lo expuesto en este apartado, se puede concluir que los Modelos Ocultos de Markov son herramientas excelentes para tareas de reconocimiento con señales de voz, pertenecen al estado de arte, son flexibles en cuanto a las formas en

que se pueden implementar y se pueden combinar con otros clasificadores. Por lo tanto se eligieron estos modelos para realizar el trabajo de esta tesis.

3.3 LOS MODELOS OCULTOS DE MARKOV

A partir de este apartado en adelante se referirá a los Modelos Ocultos de Markov simplemente como HMM (por sus siglas en inglés *Hidden Markov Models*).

En el apartado 3.1 se introdujo el concepto de secuencias de símbolos o secuencias de vectores para representar la señal de voz, en este aparatado y posteriores se explicará la relación de estas secuencias con los HMM, también se explicará por que los HMM son herramientas que modelan muy bien estas secuencias.

De forma general se clasifica a los HMM como un **modelo estadístico**, en este tipo de modelos se asume que la señal que modela puede ser caracterizada como un proceso aleatorio parametrizado; y que estos parámetros pueden ser **estimados** de manera precisa y bien definida [24].

Un HMM es un proceso **doblemente estocástico**, que tiene un proceso estocástico subyacente no observable (está oculto), pero puede ser observado a través de otros procesos estocásticos que producen la secuencia de observaciones [24].

El primer proceso estocástico se refiere a una **cadena de Markov**, que a grandes rasgos se puede considerar como una **máquina de estados finita** cuyas transiciones entre estados son probabilísticas; por lo tanto las transiciones son independientes del tiempo. La figura 3.1 muestra un ejemplo de cadena de Markov.

El resultado de este primer proceso estocástico es un **conjunto de estados en cada instante de tiempo**, donde cada estado corresponde a un evento físico observable. Este conjunto de estados es el que se encuentra “oculto”.

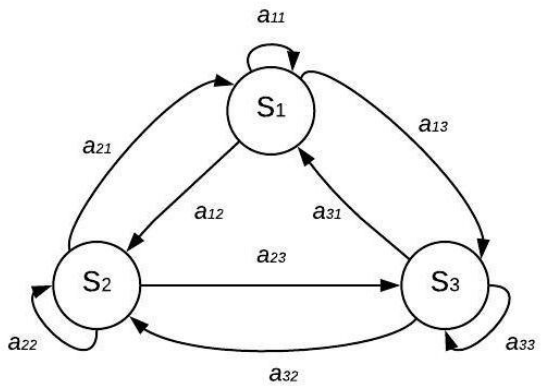


Fig. 3.1. Cadena de Markov de tres estados, donde a_{ij} es una probabilidad de transición del estado S_i al estado S_j.

El segundo proceso estocástico involucra a las observaciones resultantes del primer proceso, en este caso dichas observaciones son una **función probabilística** del estado, es decir, existe una probabilidad que una observación determinada provenga de un estado determinado. La figura 3.2 ilustra el concepto de HMM de manera general.

En [24] se puede encontrar un par de ejemplos muy sencillos que ilustran los conceptos de cadena de Markov y HMM.

En los siguientes apartados se explicará con más detalle otros conceptos relacionados con HMM, así como los parámetros que los caracterizan y la forma de estimar los mismos.

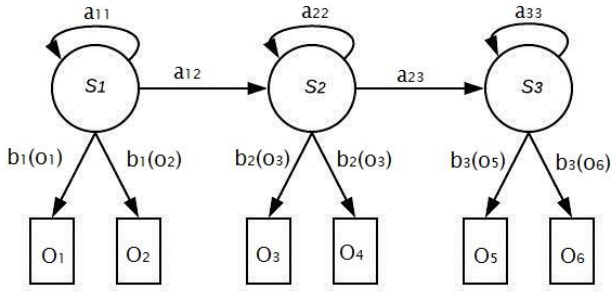


Fig. 3.2. Modelo Oculto de Markov de tres estados, donde $b_i(O_k)$ es una función probabilística de generar la observación O_k.

3.3.1 ELEMENTOS DE LOS MODELOS OCULTOS DE MARKOV

Para describir los elementos que conforman los HMM se seguirá la nomenclatura presentada en [24] ya que es la más utilizada en trabajos publicados sobre HMM.

1. *N*, es el número total de estados que componen el HMM. Los estados individuales se denotan como $S = \{S_1, S_2, \ldots, S_N\}$, y al referirse a un estado en un tiempo *t* se denota como q_t.

2. *M*, es el número de símbolos observables distintos entre sí (este elemento se considera solo si el número de símbolos es finito). Cada símbolo individual se denota como $V = v_1, v_2, \ldots, v_M$; donde *T* es el número de observaciones distintas que pueden resultar.

3. La distribución de probabilidad de las transiciones entre estados $A = \{a_{ij}\}$ donde

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i]. \quad (12)$$

Se tiene que $1 \leq i, j \leq N$, la ecuación (12) describe la probabilidad condicional de que el estado en el tiempo *t+1* sea S_j dado que el estado en *t* es S_i, esta probabilidad indica que la transición no depende del tiempo sino de los estados presente y futuro.

A es una matriz de probabilidades de transición, para un HMM como el de la figura 3.2 la matriz queda como sigue:

$$A = \begin{matrix} a_{11} & a_{12} & 0 \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{matrix}$$ Si la transición no existe la probabilidad se iguala a 0.

Hay que tomar en cuenta que los coeficientes de *A* tienen la siguiente propiedad: $\sum_{j}^{N} a_{ij} = 1$, esta propiedad indica que la suma de las probabilidades de transición del estado S_i a todos los estados S_j es igual a 1.

4. La distribución de probabilidad de que el resultado sea una observación generada por el estado S_j, $B = \{b_j(k)\}$ donde:

$$b_j(k) = P[v_k \text{ sea observado en } t | q_t = S_j]. \quad (13)$$

Se tiene que $1 \leq j \leq N$ y $1 \leq k \leq M$, la ecuación (13) describe la probabilidad condicional de que se haya observado el símbolo v_k en el tiempo *t* dado que el estado es S_j.

Al igual que a_{ij}, este parámetro también cumple con la propiedad: $\sum_{k=1}^{M} b_j\ k\ = 1$. La suma de las probabilidades de generar las observaciones de los símbolos de *V* en el estado S_j.

5. La distribución de estado inicial $\pi = \{ \pi_i \}$ donde

$$\pi_i = P[q_1 = S_i]. \quad (14)$$

Se tiene que $1 \leq i \leq N$, la ecuación 14 describe la probabilidad de que el primer estado de una secuencia sea S_i.

También el parámetro π tiene la propiedad: $\sum_{i=1}^{N} \pi_i = 1$. La suma de los valores de la distribución de estado inicial es igual a uno.

Tomando en cuenta todos estos elementos, un HMM puede generar una secuencia de observaciones $O = O_1, O_2, \ldots, O_T$; cada observación O_t es un símbolo del vector *V* (en el caso de símbolos finitos) y *T* es el número total de observaciones de la secuencia.

En muchos trabajos sobre HMM se utiliza la notación que se observa en la ecuación (15):

$$\lambda = (A, B, \pi), \quad (15)$$

donde λ es un HMM con parámetros *A*, *B* y π.

El proceso con el cual un HMM genera una secuencia de observaciones se presenta a continuación:

1. Se elige un estado inicial $q_1 = S_i$ de acuerdo a la distribución de estado inicial π.

2. Iniciando en $t = 1$, se elige $O_t = v_k$ de acuerdo a la distribución de probabilidad $b_j(k)$ del estado S_i.

3. Se hace la transición a otro estado $q_{t+1} = S_j$ de acuerdo con la distribución de probabilidad de transición del estado S_i, es decir, a_{ij}.

4. Se continua en $t = t + 1$, se repite desde la segunda parte del paso 2 hacia adelante, hasta que $t = T$.

En un HMM los elementos N y M determinan su estructura, mientras que A, B y π son los parámetros que se deben estimar.

3.3.2 LOS TRES PROBLEMAS BÁSICOS DE LOS MODELOS OCULTOS DE MARKOV

Es importante conocer los conceptos de los **tres problemas de los HMM**, estos problemas están directamente relacionados con la estimación de parámetros y la generación de las observaciones [24][34].

1. **El problema de la evaluación:** Dada una secuencia de observaciones $O = O_1, O_2,\ldots, O_T$ y un modelo $\lambda = (A, B, \pi)$, calcular eficientemente $P(O|\lambda)$, que es la probabilidad de que un modelo λ haya generado la secuencia O.

2. **El problema de la decodificación:** Dada una secuencia de observaciones O y un modelo λ, seleccionar la secuencia de estados $Q = q_1, q_2,\ldots, q_T$ con la mayor probabilidad de producir la secuencia de observaciones.

3. **El problema del entrenamiento:** Ajustar los parámetros del modelo λ, para hallar el valor máximo de $P(O|\lambda)$.

En los problemas 1 y 2 los parámetros se conocen, pero en el tercer problema el objetivo es estimar los valores de los parámetros.

3.3.2.1 SOLUCIÓN AL PROBLEMA 1

Este problema es la evaluación de lo bien que un modelo se compara con una secuencia de observaciones. Este punto de vista es importante en caso de tener varios modelos para una secuencia de observaciones.

En [24] y [34] se expone la dificultad que representa el cálculo de los componentes de A y B mediante una aproximación probabilística simple, la dificultad consiste en que el número de cálculos se eleva hasta el punto en que no es posible realizarlos de manera eficiente.

Sin embargo, para resolver este problema se creó un algoritmo conocido como **avance-retroceso** (en inglés es conocido como *forward-backward*). El algoritmo se divide en dos partes: la primera parte es avance y la segunda es el retroceso.

La parte de **avance** es la única que se necesita para resolver el problema 1, esta parte requiere de la variable $\alpha_t(i)$ que se define como:

$$\alpha_t\ i\ = P(O_1 O_2 \ldots O_t, q_t = S_i|\lambda). \tag{16}$$

Esta variable es la probabilidad de que se genere la secuencia parcial de observaciones, $O_1\ O_2 \ldots O_t$, y se esté en el estado S_i en el tiempo t, dado el modelo λ.

Para resolver $\alpha_t(i)$ se procede:

1) Inicialización:

$$\alpha_1\ i\ = \pi_i b_i(O_1), \tag{17}$$

donde $1 \leq i \leq N$.

2) Inducción:

$$\alpha_{t+1}\ j\ = \ \sum_{i=1}^{N} \alpha_t\ i\ a_{ij}\ b_j(O_{t+1}), \tag{18}$$

donde $1 \leq j \leq N$ y $1 \leq t \leq T\text{-}1$.

3) Terminación:

$$P\ O\ \lambda\ = \ \sum_{i=1}^{N} \alpha_T(i). \tag{19}$$

El objetivo del algoritmo es tomar en cuenta todas las secuencias de estados posibles para calcular $P(O|\lambda)$. La figura 3.3 ilustra las operaciones realizadas por esta primera parte del algoritmo. El inciso a) demuestra una transición desde uno de los N estados hacia un estado S_j, en el inciso b) se observa una red que describe todas las posibles secuencias de estados desde $t = 1$ hasta $t = T$ para un modelo dado.

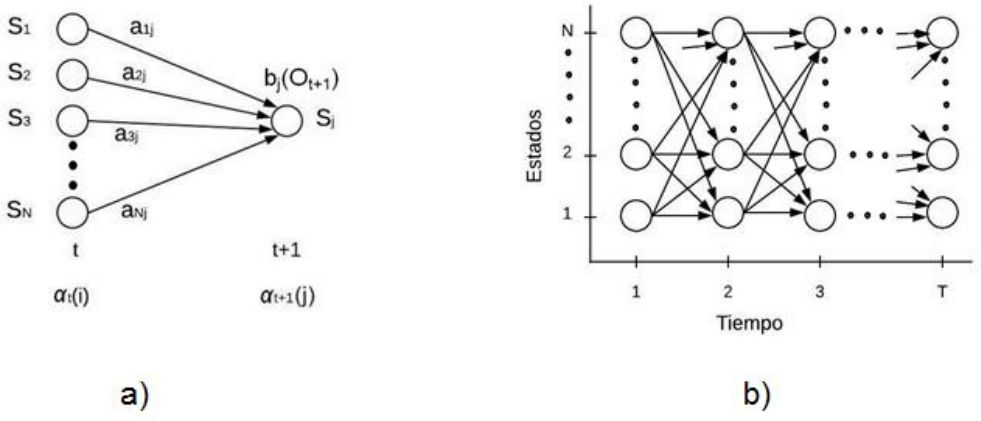


Fig. 3.3. a) Transición de estados del algoritmo de avance. b) Red de secuencias de estados que representa todas las operaciones del algoritmo de avance [24].

A pesar de que el problema 1 ya está resuelto con la parte de avance del algoritmo, a continuación se explicará lo relacionado con la parte de retroceso, ya que se utilizará para resolver el problema 3.

La justificación para la parte de **retroceso** es que la variable $\alpha_t\ (i)$ solo toma en cuenta una parte de la secuencia de observaciones desde O_1 hasta O_t, sin embargo, es necesaria una variable que tome en cuenta la información que se encuentra en el resto de la secuencia de observaciones, desde O_{t+1} hasta O_T.

Para iniciar el retroceso se define la variable $\beta_t\ (i)$ con la siguiente ecuación:

$$\beta_t\ i\ = P(O_{t+1}O_{t+2}\dots O_T|q_t = S_i, \lambda). \tag{20}$$

Esta variable es la probabilidad de que se genere la secuencia de observaciones desde $t + 1$ hasta T, dado que el estado actual es S_i y se tiene el modelo λ.

Para resolver $\beta_t\ (i)$ se procede de la siguiente manera:

1) Inicialización:

$$\beta_T\ i\ = 1, \tag{21}$$

con $1 \le i \le N$.

2) Inducción:

$$\beta_t\ i\ = \ \sum_{j=1}^{N} a_{ij} b_j(O_{t+1})\beta_{t+1}(j), \tag{22}$$

donde $1 \le i \le N$ y $t = T - 1, T - 2, \dots, 1$.

3) Adicionalmente se tiene el siguiente resultado si se efectúan las operaciones hasta $t = 1$:

$$P\ O\ \lambda\ = \ \sum_{i=1}^{N} \pi_i b_i(O_1)\beta_1(i). \tag{23}$$

La parte del retroceso funciona de manera similar a la primera parte, con la obvia excepción de que se inicia desde la última observación y se retrocede en el tiempo, la figura 3.4 ilustra los cálculos que se realizan, análogamente la red de la figura 3.3b) también se puede aplicar a este caso, tan solo se debe invertir el sentido de las flechas.

Los procedimientos de avance y retroceso deben entregar el mismo valor de $P(O|\lambda)$ porque están calculando la misma probabilidad [34].

Como se mencionó anteriormente ambos parámetros $\alpha_t\ (i)$ y $\beta_t\ (i)$ servirán para resolver el problema 3.

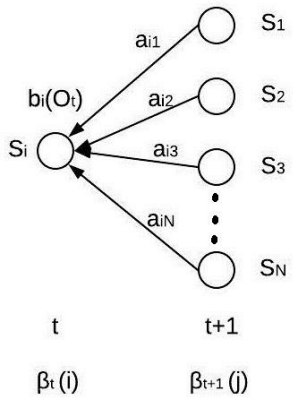


Fig. 3.4. Transición de estados en el algoritmo de retroceso [24].

3.3.2.2 SOLUCIÓN AL PROBLEMA 2

El problema 2 consiste en encontrar la secuencia de estados óptima para una secuencia de observaciones dada.

En [24] y [34] se expone que es posible resolver este problema. Se define la variable $\gamma_t(i)$ de acuerdo a la siguiente ecuación:

$$\gamma_t(i) = P(q_t = S_i | O, \lambda). \tag{24}$$

La ecuación (24) es la probabilidad de estar en el estado S_i en el tiempo t, dada la secuencia de observaciones O y el modelo λ. Esta misma ecuación se puede expresar en términos de las variables $\alpha_t(i)$ y $\beta_t(i)$:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}. \tag{25}$$

Para obtener el estado más probable q_t en el tiempo t se tiene:

$$q_t = \underset{1 \le i \le N}{argmax}[\gamma_t(i)]. \tag{26}$$

La ecuación (26) toma el estado más probable para cada tiempo t.

El problema con esta aproximación es que existe la posibilidad de que dentro de la secuencia más probable de estados, uno de ellos o la secuencia completa no sean válidos.

Para resolver el problema anterior se utiliza el **algoritmo Viterbi**, el cual define la siguiente variable:

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P[q_1 q_2 \ldots q_{t-1}, O_1 O_2 \ldots O_t, q_t = S_i | \lambda]. \tag{27}$$

La ecuación (27) indica el valor de probabilidad más alto de que en una secuencia parcial de estados, el estado en el tiempo *t* el estado sea S_i, con una secuencia de observaciones parcial hasta el tiempo *t*, dado un modelo λ.

Para calcular el valor de $\delta_{t+1}(j)$ tenemos:

$$\delta_{t+1}\ j\ = [\max_{i} \delta_t\ i\ a_{ij}] b_j(O_{t+1}). \tag{28}$$

Para generar la secuencia, se almacena el resultado de la ecuación 28 para cada instante de tiempo *t* y para cada valor de *j*. Con este propósito se tiene el arreglo $\varphi_t(j)$.

Para realizar el algoritmo Viterbi se tiene el siguiente procedimiento:

1) Inicialización:

$$\delta_1\ i\ = \pi_i b_i(O_1), \tag{29}$$

$$\varphi_1\ i\ = 0, \tag{30}$$

donde $1 \le i \le N$.

2) Recursión:

$$\delta_t\ j\ = \max_{1 \le i \le N} [\delta_{t-1}\ i\ a_{ij}] b_j(O_1), \tag{31}$$

$$\varphi_t\ j\ = \operatorname*{argmax}_{1 \le j \le N} [\delta_{t-1}\ i\ a_{ij}], \tag{32}$$

donde $1 \le j \le N$ y $2 \le t \le T$.

3) Terminación:

$$q_T = \operatorname*{argmax}_{1 \le i \le N} [\delta_T\ i\ ]\ . \tag{33}$$

4) Se recupera la secuencia de estados retrocediendo:

$$q_t = \varphi_{t+1}(q_{t+1}), \tag{34}$$

donde $t = T - 1, T - 2, \ldots, 1$.

El algoritmo Viterbi es similar a la parte de avance del algoritmo avance-retroceso, sin incluir la parte de retroceso. La diferencia más importante es el hecho de que el algoritmo Viterbi elige el valor máximo en cada tiempo *t*, mientras que en parte de avance se suman todos los resultados de los cálculos en cada tiempo *t*. El algoritmo Viterbi también puede ser descrito con una red como la que se encuentra en la figura 3.3b) respetando la diferencia anterior.

3.3.2.3 SOLUCIÓN AL PROBLEMA 3

El tercer problema de los HMM es estimar el valor de los parámetros *(A, B* y *π)* para maximizar la probabilidad de que una secuencia de observaciones sea generada por un modelo *P(O|λ)* [24].

No hay una forma analítica ni óptima de estimar los parámetros del HMM. Existen varias técnicas para hacer la estimación de los parámetros y maximizar *P(O|λ),* sin embargo, la más utilizada es el **método Baum-Welch.**

Baum-Welch es una técnica de estimación iterativa, lo que significa que en cada iteración los parámetros son reestimados y mejorados, lo que hace que *P(O|λ)* sea maximizado localmente. Dada una secuencia o secuencias de observación como datos de entrenamiento y un conjunto de parámetros cuyos valores sean inicializados aleatoriamente, se reestiman en cada iteración de tal forma que se cumple lo siguiente:

$$P(O|\lambda) \leq P(O|\lambda), \tag{35}$$

donde $\lambda = (A, B, \pi)$ es el modelo reestimado a partir de $\lambda = (A, B, \pi)$, esta condición implica una mejora de los parámetros y la maximización de *P* con cada iteración. La relación expresada en (35) indica implícitamente la condición de terminación del método Baum-Welch, si el valor de $P(O|\lambda)$ no es mayor a $P(O|\lambda)$ la reestimación se detiene.

Primero definimos la variable $\xi_t\ (i,j)$ como la probabilidad de estar en el estado S_i en el tiempo *t* y en el estado S_j en *t + 1*, dados un modelo y una secuencia de observaciones:

$$\xi_t\ i,j\ = P(q_t = S_i, q_{t+1} = S_j|O, \lambda). \tag{36}$$

Se puede escribir $\xi_t\ (i,j)$ en términos las variables del algoritmo avance-retroceso:

$$\xi_t\ i,j\ = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)} = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}, \tag{37}$$

donde el numerador es la ecuación (36) en términos del algoritmo avance-retroceso y el denominador permite que el resultado sea un valor de probabilidad [24] [34].

En el apartado 3.3.2.2 se presentó la variable $\gamma_t\ (i)$, podemos escribir $\gamma_t\ (i)$ en términos de $\xi_t\ (i,j)$ haciendo una sumatoria en el índice *j*.

$$\gamma_t\ i\ = \ \sum_{j=1}^{N}\xi_t(i,j). \tag{38}$$

Si se realiza la sumatoria de $\gamma_t\ (i)$ en *t* obtenemos el número esperado de veces en el que se visitó el estado S_i, esto es:

$$\sum_{t=1}^{T-1}\gamma_t(i) = número\ esperado\ de\ veces\ que\ se\ visitó\ el\ estado\ S_i\ . \tag{39}$$

También se puede interpretar la sumatoria de (39) como el número esperado de veces que se hizo una transición desde el estado S_i hasta el tiempo *T - 1*.

Igualmente, si hacemos la sumatoria en el tiempo hasta *T - 1* para la variable $\xi_t(i,j)$ se tiene el número esperado de veces que hubo una transición del estado S_i al estado S_j:

$$\sum_{t=1}^{T-1} \xi_t(i,j) = número\ esperado\ de\ transiciones\ de\ S_i\ a\ S_j. \quad (40)$$

Si utilizamos el concepto de contar eventos en un HMM como los mencionados en (39) y (40), podemos hacer la reestimación de los parámetros de un HMM de la siguiente forma:

$$\pi_i = número\ esperado\ de\ veces\ de\ iniciar\ \ t = 1\ \ en\ el\ estado\ S_i = \gamma_1(i), \quad (41)$$

$$a_{ij} = \frac{número\ esperado\ de\ transiciones\ del\ estado\ S_i\ al\ estado\ S_j}{número\ esperado\ de\ transiciones\ que\ parten\ del\ estado\ S_i} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i,j)}, \quad (42)$$

$$b_j\ k\ = \frac{número\ esperado\ de\ veces\ que\ se\ observa\ el\ símbolo\ v_k\ estando\ en\ el\ estado\ S_j}{número\ esperado\ de\ veces\ que\ se\ visitó\ el\ estado\ S_j} = \frac{\sum_{t=1,\ si\ O_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}. \quad (43)$$

Si se realizan varias iteraciones del método Baum-Welch, se cumple la relación (35) y se detienen las reestimaciones cuando se cumpla la condición ya establecida. Los parámetros reestimados también cumplirán con las propiedades descritas en el apartado 3.3.1.

El resultado final del proceso de reestimación se conoce como **estimación de máxima verosimilitud** (por sus siglas en inglés ***MLE***).

Nota: La **función de verosimilitud** o simplemente **verosimilitud** es el término en español equivalente a *likelihood function* o simplemente *likelihood*. La verosimilitud es definida como una función de los parámetros de un modelo estadístico, dado un conjunto de información observada. El término muchas veces es usado indistintamente para referirse al concepto de probabilidad, sin embargo, en este trabajo se utilizará explícitamente el término verosimilitud con su definición correspondiente cuando sea pertinente.

3.3.3 DENSIDADES DE OBSERVACIONES CONTINUAS EN MODELOS OCULTOS DE MARKOV

En el apartado 3.3.1 se presentaron los elementos (parámetros) que forman parte de los HMM, se mencionó que el parámetro *V* es el número finito de símbolos que se pueden observar, esto implica que las observaciones son discretas, sin embargo, es necesario tomar en cuenta el caso en el que las **observaciones son continuas**.

Normalmente los vectores de observaciones continuas se discretizan para ser manejables por una computadora, pero en algunas aplicaciones con HMM se presenta una degradación en los resultados esperados si se toma la aproximación de utilizar **densidades de probabilidad discretas** para trabajar con dichas observaciones continuas [24].

Las señales de voz son continuas y los vectores de características que se extraen de ella también son considerados continuos, es por esto que los HMM deben ser implementados con **densidades de observaciones continuas**, para reconocimiento de hablante los vectores de características de la señal de voz serán las observaciones continuas [24].

Para implementar las densidades de observaciones continuas se tiene una forma de estimar los parámetros de la **función de densidad de probabilidad** (también conocida simplemente como **densidad**) de las observaciones. Para representar una densidad se utiliza un **modelo de mezclas finitas** de la forma:

$$b_j(\boldsymbol{O}) = \sum_{m=1}^{M} c_{jm}\mathfrak{N}(\boldsymbol{O}, \boldsymbol{\mu}_{jm}, \boldsymbol{U}_{jm}), \tag{44}$$

donde $1 \leq j \leq N$ y $1 \leq m \leq M$, siendo M el número de mezclas, $\boldsymbol{O}$ es el vector de observaciones que se modela, c_{jm} es el peso o ganancia de una mezcla m en el estado S_j y $\mathfrak{N}$ es una densidad cóncava o elípticamente simétrica [24], con un vector de **medias** $\boldsymbol{\mu}_{jm}$ y **matriz de covarianza** $\boldsymbol{U}_{jm}$ para la mezcla m en el estado S_j. Un ejemplo de mezcla se encuentra en la figura 3.5, en ella hay tres campanas gaussianas (cada campana gaussiana es un componente) que *mezcladas* forman una nueva densidad.

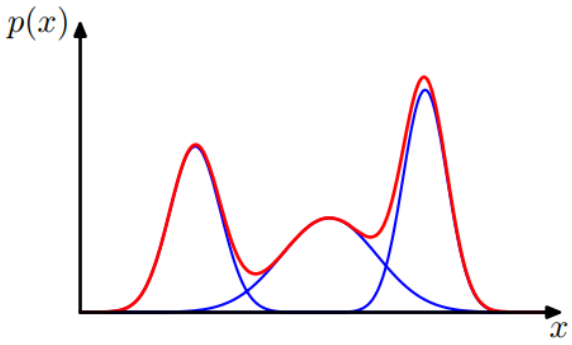


Fig. 3.5. Ejemplo de modelo de mezclas finitas compuesto por tres campanas gaussianas.

Los pesos de las mezclas c_{jm} satisfacen las siguientes condiciones: $\sum_{m=1}^{M} c_{jm}$ y $c_{jm} \geq 0$.

Las ecuaciones del método Baum-Welch para reestimar los parámetros c_{jm}, $\boldsymbol{\mu}_{jm}$ y $\boldsymbol{U}_{jm}$ son las siguientes:

$$c_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j,k)}, \tag{45}$$

$$\boldsymbol{\mu}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)\boldsymbol{o}_t}{\sum_{t=1}^{T} \gamma_t(j,k)}, \tag{46}$$

$$U_{jk} = \frac{\sum_{t=1}^{T} \gamma_t\ j,k\ (o_t - \mu_{jk})(o_t - \mu_{jk})^T}{\sum_{t=1}^{T} \gamma_t(j,k)}, \tag{47}$$

donde $\gamma_t\ (j,k)$ es la probabilidad de estar en el estado S_j en el tiempo t y que el componente k de la mezcla haya generado la observación $\boldsymbol{O}_t$. Esta probabilidad tiene la siguiente ecuación:

$$\gamma_t\ j,k\ = \frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}\ \ \frac{c_{jk}\Re(o_t,\mu_{jk},U_{jk})}{\sum_{m=1}^{M} c_{jm}\Re(o_t,\mu_{jm},U_{jm})}\ . \tag{48}$$

Es fácil relacionar la ecuación (48) con la definición dada en las ecuaciones (24) y (25), $\gamma_t\ (j)$ es una generalización de $\gamma_t\ (j,k)$, siendo el primero el caso en el que se tiene una sola mezcla en el estado S_j o una densidad discreta.

La ecuación (45) para reestimar c_{jk} es la tasa entre la cantidad esperada de veces que el sistema está en el estado S_j usando el componente k de la mezcla y el número esperado de veces en que el sistema está en el estado S_j.

En las ecuaciones (46) y (47) cada vector de observaciones $\boldsymbol{O}_t$ contribuye al cálculo de los valores de máxima verosimilitud para cada estado S_j. Esto significa que se considera que cada estado genera cada vector de observaciones en proporción a la probabilidad de estar en ese estado cuando el vector fue observado [36].

3.3.4 MODELO BAKIS

Hasta este punto los conceptos expuestos son aplicables para HMM completamente conectados (se puede hacer la transición de un estado hacia cualquier otro estado) que es un ejemplo de los llamados **modelos ergódicos**, la figura 3.1 es un ejemplo de estos modelos.

Un tipo de HMM muy utilizado es el **modelo Bakis** o **modelo izquierda-derecha**, este tiene una secuencia de estados cuya propiedad más importante es que con el paso del tiempo el índice de los estados incrementa [24], este modelo se ilustra en la figura 3.6.

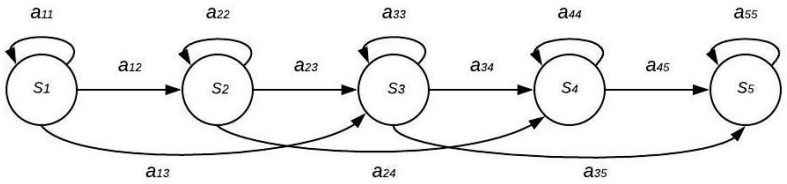


Fig. 3.6. Modelo Bakis o izquierda-derecha.

Las transiciones de este modelo tienen la siguiente propiedad: $a_{ij} = 0$ si $j < i$, lo que significa que no se permiten transiciones a estados cuyo índice sea menor al del estado actual, tampoco se permite hacer un "salto" de más de dos estados; y los estados primero

y último no tienen transiciones entre sí. La matriz de transiciones *A* es similar a la presentada en el apartado 3.3.1, cuyo tamaño dependerá del número de estados.

Otra propiedad interesante es que:

$$\pi_i = \begin{matrix} 0 \; si \; i \neq 1 \\ 1 \; si \; i = 1 \end{matrix}. \tag{49}$$

Esta propiedad es obvia dado que la secuencia de estados debe iniciar en el estado 1 y terminar en el estado *N*.

Las características y propiedades del modelo izquierda-derecha no afectan demasiado al proceso de reestimación de los parámetros.

También tiene la propiedad de modelar señales cuyas propiedades cambian con el tiempo, por ejemplo la señal de voz [24], este hecho será más relevante posteriormente.

3.3.4.1 REESTIMACIONES DE PARÁMETROS CON MÚLTIPLES SECUENCIAS DE OBSERVACIONES

Debido a las propiedades de los HMM izquierda-derecha, no pueden ser entrenados con una sola secuencia de observaciones. Esto es debido a que la naturaleza transitoria de los estados solo permite un pequeño número de observaciones para un estado (hasta que haya una transición a otro estado) [24].

Por lo tanto se hace una modificación al proceso de reestimación para tomar en cuenta varias secuencias de observación como la siguiente:

$$\boldsymbol{O} = [\boldsymbol{O}^1, \boldsymbol{O}^2, \boldsymbol{O}^3, \dots, \boldsymbol{O}^R], \tag{50}$$

donde $\boldsymbol{O}^{(r)} = [O_1{}^r, O_2{}^r, O_3{}^r, \dots, O_{T_r}{}^r]$ es la secuencia de observación *r*. Se asume que las secuencias de observaciones son independientes entre sí y la maximización de *P(O|λ)* se reescribe como:

$$P\ O\ \lambda\ =\ \prod_{r=1}^{R} P\ \boldsymbol{O}^{\,r}\ \lambda\ =\ \prod_{r=1}^{R} P_r. \tag{51}$$

Las ecuaciones de reestimación que consideran múltiples secuencias de observaciones son:

$$a_{ij} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r-1} \alpha_t^r(i) a_{ij} b_j(O_{t+1}^r) \beta_{t+1}^r(i)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r-1} \alpha_t^r(i) \beta_t^r(i)}. \tag{52}$$

A manera de complemento, también se presenta la reestimación de $b_j(\ell)$ para el caso de secuencias de observaciones discretas:

$$b_j\ \ell\ = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{\substack{t=1 \\ con\ O_t = v_\ell}}^{T_r-1} \alpha_t^r(i) \beta_t^r(i)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r-1} \alpha_t^r(i) \beta_t^r(i)}. \tag{53}$$

El parámetro π_i no es necesario reestimarlo dadas las condiciones de la relación dada en (49).

3.4 EL CONJUNTO DE HERRAMIENTAS HTK (*HIDDEN MARKOV MODEL TOOLKIT*)

En todo el apartado 3.3 se expusieron los detalles concernientes a los HMM, desde su concepto hasta la forma de estimar sus parámetros teniendo una secuencia de vectores de observaciones, que pueden ser continuos o discretos. Pero para realizar dichas estimaciones es necesario trasladar todas las ecuaciones y condiciones a un algoritmo implementado en software, esta aproximación no es el objetivo de este trabajo, es por esto que se escogió un conjunto de herramientas de software que maneje los cálculos y esté completamente orientado al entrenamiento y evaluación de HMM.

Las herramientas de software escogidas fueron las incluidas en HTK, cabe aclarar que no es una aplicación como MATLAB, por el contrario son herramientas individuales que deben ser coordinadas para realizar la tarea de entrenamiento y evaluación de HMM, así como el manejo de tareas secundarias como administración de archivos necesarios para estos propósitos.

HTK está orientado principalmente a HMM para procesamiento y reconocimiento de voz, aunque es posible realizar tareas para otras áreas en las que se utilizan HMM, es necesario darle el enfoque que se dirija a reconocimiento de hablante.

En posteriores apartados se explicarán las generalidades de HTK que son relevantes a este trabajo sin entrar en muchos detalles, para mayor información sobre el funcionamiento de HTK se puede consultar [36].

3.4.1 GENERALIDADES DE HTK

Como ya se mencionó HTK es un conjunto de herramientas cuyo propósito es el entrenamiento y evaluación de HMM, así que es necesario conocer las suposiciones que hace y condiciones bajo las cuales realiza las tareas con HMM.

La primera condición de HTK es que solo trabaja HMM de izquierda a derecha. Toma en cuenta las propiedades de estos modelos y agrega otras condiciones: los estados S_1 y S_N no tienen transición hacia sí mismos y no emiten ninguna observación, es decir, $b_1(O) = 0$ y $b_N(O) = 0$; el motivo de esto es que estos estados sirven como conexión entre diferentes HMM para el caso en que se hace reconocimiento continuo de voz; no está permitido el "salto" de hasta 2 estados.

Otra condición es que los HMM tienen densidades continuas, cada estado tiene asociada una sola mezcla finita de *M* componentes, la mezcla finita está compuesta por **gaussianas multivariadas**, cada una tiene la siguiente ecuación:

$$\mathcal{N}(\boldsymbol{O}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^n |\Sigma|} e^{-\frac{1}{2}(\boldsymbol{O}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{O}-\boldsymbol{\mu})}, \quad (54)$$

donde $\boldsymbol{\mu}$ es el vector de medias y matriz de covarianza $\boldsymbol{\Sigma}$, *n* es el número de dimensiones de $\boldsymbol{O}$.

Con respecto a las observaciones para entrenar el HMM, estas se dividen en **flujos de datos**, sin entrar en muchos detalles al respecto, en el caso de este trabajo los vectores de observaciones serán los MFCC los cuales son considerados el primer flujo, mientras que los coeficientes delta y delta-delta son el segundo y tercer flujo respectivamente. Con esta información la ecuación para estimar $b_j(\boldsymbol{O}_t)$ se define:

$$b_j(\boldsymbol{O}_t) = \prod_{s=1}^{S} \left[\sum_{m=1}^{M_s} c_{jsm} \mathcal{N}(\boldsymbol{O}_{st}, \boldsymbol{\mu}_{jsm}, \boldsymbol{\Sigma}_{jsm}) \right]^{\gamma_s}, \tag{55}$$

donde S es el número de flujos de datos independientes, M_s es el número de componentes de la mezcla de gaussianas del flujo s y $\acute{Y}_s$ es el peso de un flujo de datos (no confundir con $\gamma_t(j)$).

La figura 3.7 muestra los detalles de un HMM con el que las herramientas de HTK trabajan.

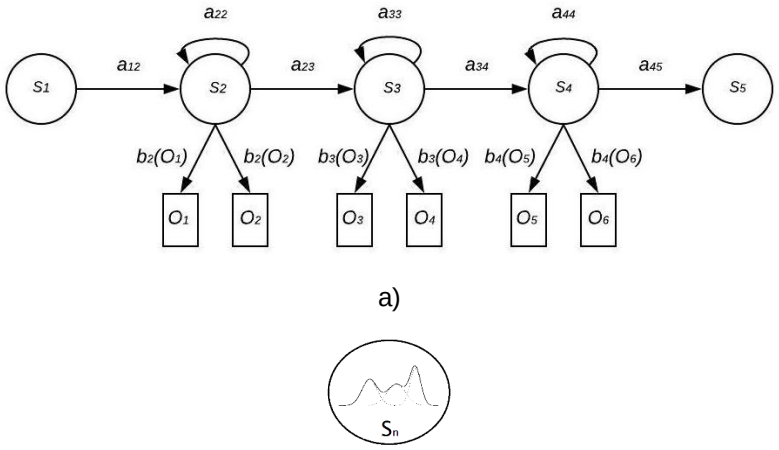


Fig. 3.7. a) Modelo Oculto de Markov de HTK. b) Ejemplo de estado S_n con una mezcla de gaussianas [36].

Las herramientas de HTK utilizan el método de reestimación Baum-Welch y el algoritmo Viterbi como se describieron en los apartados correspondientes a las soluciones de los problemas de HMM, así como la ecuación de reestimación de a_{ij} para el caso de tener múltiples secuencias de observaciones, aunque por razones de cómputo, todos los cálculos se realizan en el dominio logarítmico.

Cabe mencionar que la implementación de estos procedimientos en HTK puede incluir otros planteamientos adicionales a los ya expuestos que se pueden consultar en [36]. Uno de estos planteamientos es la implementación de un algoritmo conocido como **token passing** para el cálculo de $\log P(O|\lambda)$, HTK hace una normalización dividiendo la probabilidad logarítmica entre el número de segmentos en que se dividió la señal de voz para el cálculo de los vectores de características, más detalles de este planteamiento se pueden consultar [36].

En el capítulo 5 se expondrá con detalle el uso de las herramientas de HTK en conjunción con otros paquetes de software.

3.5 MODELANDO A LOS HABLANTES

En el apartado 1.7.1 del capítulo 1 y en el apartado 3.1 de este capítulo se introdujo el concepto de **modelo de hablante**, en este apartado y sus subsecciones se profundizarán los conceptos sobre este y otro tipo de modelo que es necesario para la tarea de verificación de hablante, además se explicará la motivación para construir modelos que representen a los hablantes.

3.5.1 LA RAZÓN DE VEROSIMILITUD

Los modelos proveen un método para calificar la comparación de una muestra de prueba con el modelo mismo [4]. En verificación de hablante existe un modo estándar para hacer esta calificación y está completamente relacionado con los modelos de hablantes: **la razón de verosimilitud** (***likelihood ratio*** en inglés) [25].

Para comprender esta razón primero tomaremos en cuenta lo siguiente: dado un segmento de voz *Y*; y un hablante hipotético *s*, la verificación de hablante consiste en determinar si *Y* fue hablado por *s* (asumiendo que *Y* solo fue pronunciado por un solo hablante).

La tarea de verificación puede ser planteada como la competencia entre dos hipótesis:

H_0: *Y* fue pronunciado por el hablante hipotético *s*.

H_1: *Y* **no** fue pronunciado por el hablante hipotético *s*.

Para decidir una de estas dos hipótesis se tiene la prueba de la razón de verosimilitud dada por:

$$LR\ X\ = \frac{p(Y|H_0)}{p(Y|H_1)} \quad \begin{matrix} \geq\ \theta\ aceptar\ H_0 \\ <\ \theta\ rechazar\ H_0 \end{matrix}, \tag{56}$$

donde $p(Y|H_i)$, con $i = 0, 1$; es la función de verosimilitud de que *Y* haya sido generada por H_i. Mientras que θ es un umbral de decisión para aceptar o rechazar H_0. Para un sistema de verificación de hablante el objetivo básico es crear técnicas para calcular valores para ambas funciones de verosimilitud.

Matemáticamente hablando H_0 es un modelo denotado como λ_s que representa las características acústicas del hablante hipotético *s* (el modelo de hablante), mientras que *Y* se denota como *X*, donde $X = \{x_1, x_2, \ldots, x_T\}$, es un conjunto de *vectores de características* extraídos de un segmento de voz. Por otro lado la hipótesis H_1 se denota como el modelo λ, que representa al espacio de posibles alternativas al hablante hipotético λ_s.

Casi siempre se utiliza el logaritmo base 10 de estas funciones de verosimilitud obteniendo la **razón de verosimilitud logarítmica** (***log-likelihood ratio*** en inglés) la cual se escribe como sigue:

$$\Lambda\ X\ = \log p\ X\ \lambda_s\ - \log\ p(X|\lambda). \tag{57}$$

La razón expresada en la ecuación (57) será la evaluación utilizada en este trabajo para la tarea de verificación de hablante, adicionalmente, a partir de este punto se referirá a esta razón por sus siglas en inglés ***LLR*** por cuestiones de simplicidad.

Algunas nociones del modelo λ_s ya han sido expresadas hasta este punto, por otro lado el modelo λ puede ser definido mediante dos aproximaciones. Las dos aproximaciones clásicas son ***los* modelos de cohorte** (***cohort models***) y los **modelos de hablantes globales** (***universal background models*** en inglés), en [3] y [25] se encuentran detalles sobre los primeros modelos, en este trabajo el enfoque es sobre los modelos de hablantes globales o modelos globales.

Las hipótesis que se manejan para obtener el LLR nos permiten dividir el problema de verificación en dos partes muy importantes, cada una de ellas requiere de un modelo especializado para obtener un resultado y tomar una decisión, esta es la principal motivación para tomar en cuenta más de un tipo de modelo. La figura 3.8 ilustra la verificación de hablante mediante el LLR.

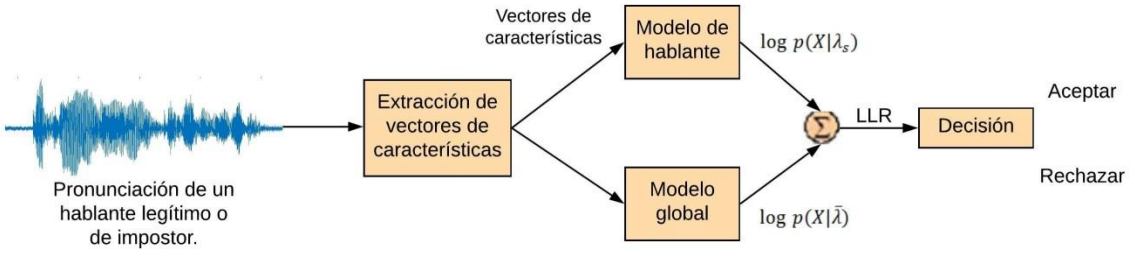


Fig. 3.8. Verificación de hablante con LLR [25].

3.5.2 MODELOS GLOBALES

Los modelos globales responden a la necesidad de tener una hipótesis alternativa que puede ser vista como un caso contrario.

Estos modelos son **independientes de los hablantes**, es decir, son entrenados utilizando una gran cantidad de información tomada de muchos hablantes que **no** están dentro del grupo de hablantes cuyas identidades se desean verificar. A estos modelos se les considera un hablante que representa el **promedio de las características acústicas** de una base de datos [25].

Para entrenar estos modelos hay distintos criterios que se pueden seguir, por ejemplo, se puede entrenar un modelo global independiente del género usando muestras de hombres y mujeres equitativamente, si se tiene la certeza del género del hablante a

quien pertenece una muestra de prueba entonces se entrenan modelos dependientes del género, aunque rara vez se tiene de antemano esta certeza; también se puede considerar separar las muestras de entrenamiento por grupos de acuerdo al hardware utilizado para grabar. En general no hay una forma establecida o recomendada de determinar la cantidad de información para el entrenamiento o el número de hablantes de los cuales tomar dicha información.

Otra función muy importante que cumplen los modelos globales tiene que ver con la riqueza acústica con la que fueron entrenados, ya que a partir de ellos se pueden **adaptar** los modelos de hablantes que retendrán esta riqueza, además de contener información de un hablante particular. Esta adaptación se verá en los siguientes apartados.

A partir de ahora estos modelos se denotarán como λ_{UBM} en sustitución a la notación usada en la ecuación (57) para la hipótesis $H_1(\lambda)$.

3.5.3 MODELOS DE HABLANTES

Los modelos de hablantes representan las características de los **hablantes legítimos** o **genuinos** [25].

Teóricamente estos modelos se entrenan utilizando la información obtenida en la etapa de enrolamiento del sistema de un hablante genuino. Sin embargo, no siempre se tienen suficientes datos de un hablante legítimo para el entrenamiento de un modelo, ya que las sesiones de enrolamiento normalmente no son suficientes para obtener suficientes datos, tampoco es viable extender las sesiones de enrolamiento porque esto no es bien recibido por los usuarios.

Como se mencionó, la alternativa más viable es tomar un modelo global que es rico en información y **adaptar** sus parámetros con los vectores de características de un hablante específico obtenidos durante el enrolamiento.

3.5.3.1 ADAPTACIÓN DE MODELOS DE HABLANTES

Las técnicas de adaptación se pueden utilizar de dos formas diferentes. Si se tienen las transcripciones de los datos de adaptación entonces el proceso se llama **adaptación supervisada**. Si no se tienen las transcripciones ni ningún tipo de datos etiquetados para la adaptación entonces este caso es conocido como **adaptación no supervisada** [36]. Si los datos de adaptación se obtienen de una sola sesión de enrolamiento, la adaptación es **estática**. Otro esquema de adaptación se puede aplicar si se adquiere más datos de adaptación con el paso del tiempo, este caso es el de **adaptación incremental** [36].

Un esquema de adaptación muy utilizado es la llamada: **adaptación bayesiana**, también conocida como **máxima estimación a posteriori** (***MAP*** por sus siglas en inglés).

Este tipo de adaptación requiere que se tenga conocimiento previo sobre los parámetros del modelo. Este conocimiento previo se refiere a utilizar los parámetros de un

modelo independiente del hablante a partir de los cuales se estimarán los parámetros para el **modelo de hablante adaptado**.

Las herramientas de HTK tienen una implementación de adaptación MAP supervisada para HMM, a partir de modelos independientes del hablante, la cual viene dada por las siguientes ecuaciones:

$$\boldsymbol{\mu}_{jm} = \frac{\mathrm{N}_{jm}}{\mathrm{N}_{jm}+\tau}\boldsymbol{\mu}_{jm} + \frac{\tau}{\mathrm{N}_{jm}+\tau}\boldsymbol{\mu}_{jm}, \tag{58}$$

donde $\boldsymbol{\mu}_{jm}$ es el vector de medias adaptado, $\boldsymbol{\mu}_{jm}$ es el vector de medias de los datos de adaptación y $\boldsymbol{\mu}_{jm}$ es el vector de medias del modelo independiente del hablante; N_{jm} es la verosimilitud de que los datos de adaptación hayan sido generados por el componente gaussiano *m* en el estado S_j, mientas que τ es el coeficiente de ponderación de los datos de adaptación.

La ecuación (58) es la adaptación del vector de medias a partir de los vectores de medias del modelo independiente del hablante y de los datos de adaptación tomando en cuenta la función de verosimilitud N_{jm}, la cual se define como:

$$\mathrm{N}_{jm} = \sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_t^r(j,m), \tag{59}$$

donde *R* es el total de vectores de observaciones como se aprecia en la ecuación (50) y $\gamma_t^r(j,m)$ tiene el mismo significado que en el apartado 3.3.3.

Por último se tiene la ecuación del vector de medias de los datos de adaptación:

$$\boldsymbol{\mu}_{jm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_t^r(j,m)\boldsymbol{o}_t^r}{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_t^r(j,m)}. \tag{60}$$

Si el valor de N_{jm} es pequeño entonces el vector de medias adaptadas permanecerá cerca al correspondiente vector del modelo independiente del hablante. Adaptación MAP actualiza cada media en el sistema basado en la ponderación de los datos de adaptación y las medias de un modelo independiente del hablante [36].

3.6 MODELADO ACÚSTICO

Anteriormente se mencionó que los modelos izquierda-derecha son excelentes para modelar las características dinámicas de la señal de voz. Otra característica muy importante de estos HMM es que pueden modelar distintos niveles del léxico con el que se esté trabajando; la elección de uno de estos niveles dependerá de las condiciones de la aplicación; a continuación se listarán los niveles más comunes que pueden modelar los HMM [4]:

- **Nivel oración***:* el uso de un solo HMM para modelar una oración es una opción cuando el léxico de los datos de enrolamiento es similar y se presenta en el mismo orden que el léxico de los datos de prueba. La aplicación de HMM a nivel oración es conocida como "*Mi voz es mi contraseña".*

- **Nivel palabra***:* si el orden en que aparece el léxico de los datos de enrolamiento y de prueba no es el mismo pero las palabras que aparecen en ambos conjuntos de datos son muy similares, entonces se pueden utilizar varios HMM para que cada uno represente una palabra del léxico. Normalmente se implementa este nivel cuando el léxico está conformado por dígitos que son agrupados en secuencias y cuyo orden dentro de la secuencia es aleatorio.

- **Nivel fonema***:* los fonemas representan la articulación mínima de un sonido vocálico o consonántico, la unión de varios fonemas conforma una palabra pronunciada, en este nivel cada HMM es entrenado para representar un solo fonema. Este nivel es particularmente útil en tareas independientes del texto ya que se tienen representado el espacio acústico del lenguaje y no es prioritario limitar el léxico.

La figura 3.8 ilustra la forma en que los HMM modelan los niveles ya mencionados.

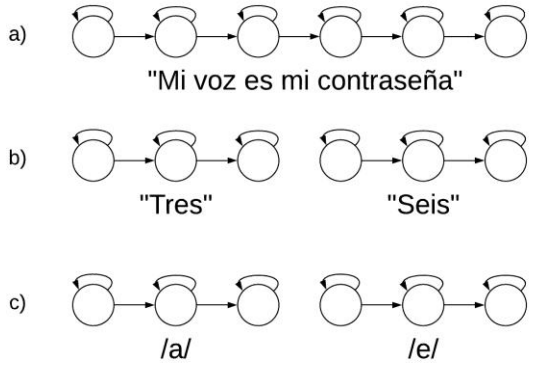


Fig. 3.9. a) HMM a nivel oración. b) HMM a nivel palabra. c) HMM a nivel fonema.

Es importante señalar que no se tienen reglas establecidas sobre las características del HMM para modelar un nivel en particular, es decir, no existe una cantidad establecida de estados ni de componentes de la mezcla de gaussianas. Aunque en muchos trabajos se han hecho propuestas para determinar las cantidades más adecuadas de acuerdo a las condiciones de la aplicación.

Para este trabajo se eligió modelar los HMM a nivel de palabra debido a las características de la base de datos que se utilizó, dichas características se expondrán detalladamente en el siguiente capítulo.

CAPÍTULO 4

LA BASE DE DATOS BIOMEX-DB

4.1 INTRODUCCIÓN

Como se expuso en el capítulo 1, un componente muy importante para hacer cualquier tipo de biometría es una base de datos que contenga ejemplos de señales generadas por el cuerpo humano de distintas personas. Estas señales de ejemplo servirán para entrenar los modelos de clasificación; en el caso de biometría de voz se entrenan los modelos globales y posteriormente los modelos de hablantes, otra función que cumplen estos datos es realizar pruebas al sistema de reconocimiento de hablante, utilizando un conjunto de muestras de prueba que simulen las condiciones de la aplicación real.

Dependiendo de la modalidad de biometría que se va a realizar y las condiciones de la aplicación, se elige una base de datos disponible públicamente o se realiza la tarea de tomar las señales para generar una base de datos que se ajuste a las necesidades de la aplicación.

Ya sea que se utilice una base de datos ya hecha o se genere una, es importante tomar en cuenta algunas consideraciones para decidir si una base de datos es adecuada para una aplicación específica:

- Revisar que las características de las señales biométricas son adecuadas para la aplicación, por ejemplo, la tasa de muestreo, la amplitud de las señales, la resolución en bits de cada muestra, el formato, etc.

- El protocolo de toma de señales, este punto se refiere a las condiciones que se dispusieron para tomar las señales, si hubo una preparación previa de los voluntarios que pueda impactar en la calidad de la señal, etc.

- Los equipos e instrumentos que se utilizaron para tomar las señales, este punto impacta directamente en la calidad de la señal biométrica.

- La población de personas que fueron voluntarias para la toma de señales biométricas, esta consideración es importante porque las características de la población como la edad, el género o la condición física pueden tener un impacto importante en los resultados.

- La cantidad de información contenida en la base de datos, esta consideración repercute en el entrenamiento de los modelos de clasificación y en los datos disponibles para realizar pruebas al sistema biométrico.

- El número de sesiones en que se tomaron los datos. Es recomendado que las señales de la base de datos sean tomadas durante varias sesiones a lo largo de un periodo no muy largo de tiempo, esto permite capturar características del voluntario que pueden variar entre sesiones.

En el ámbito de reconocimiento de hablante es necesario contar con una base de datos que contenga archivos de señales de voz. Las consideraciones expuestas anteriormente son aplicables a este tipo de base de datos, aunque también se deben considerar las siguientes condiciones:

- El léxico o vocabulario pronunciado en la base datos, si la tarea es dependiente del texto se debe tomar en cuenta las palabras y la estructura de las oraciones pronunciadas.

- El idioma de las pronunciaciones de voz. Este punto implica considerar también el acento con el que los voluntarios pronunciaron las palabras y las pronunciaciones particulares de cada hablante.

- Si las grabaciones de voz fueron realizadas en ambiente con ruido, si se realizaron con ruido en muchas ocasiones se detallan las condiciones del ambiente que generaron el ruido.

- El canal de grabación de audio, por ejemplo, el canal puede ser la línea telefónica alámbrica o inalámbrica. Este punto también conlleva considerar las condiciones del canal como el ancho de banda disponible y el ruido.

En [16] y [17] se hace un breve listado de bases de datos de voz enfocadas a tareas dependientes del texto junto con las características más destacas de cada una, muchas de ellas incluyen datos de otros rasgos biométricos.

A pesar de la gran variedad de bases de datos de voz que existen, la licencia de uso de muchas de ellas tiene un costo elevado y lamentablemente hay muy pocas bases de datos disponibles de forma gratuita, a esto hay que agregar que no se encontraron bases de datos de voz en lenguaje español que estén disponibles de manera gratuita.

Debido al inconveniente que representa el costo de la licencia de uso de bases de datos de voz populares y al hecho de que hay pocas en idioma español, se tomó la decisión de generar una base de datos en español con un protocolo propio y cuyas características se enfoquen al reconocimiento de hablante.

A la par de la toma de señales de voz también se tomaron señales EEG y de video, de las cuales las primeras servirán para un trabajo de investigación independiente sobre EEG y los archivos de video consisten en grabación del rostro de los voluntarios durante la captura de sus señales biométricas.

En este capítulo solo se expondrán los detalles de las señales que conforman la parte de voz así como el protocolo correspondiente.

4.2 LA BASE DE DATOS DE VOZ BIOMEX-DB

El objetivo de la base de datos BIOMEX-DB es tener un conjunto de señales de ejemplo para realizar distintas modalidades de biometría, la parte de voz en particular está enfocada para hacer reconocimiento de hablante dependiente del texto en idioma español, con el agregado de tener un léxico que permita utilizar contraseñas numéricas de 4 dígitos, similar al protocolo de seguridad de un cajero automático.

Esta base de datos toma varios aspectos de la base de datos de voz ***SecuVoice*** [21], aunque los protocolos de grabación son diferentes, en general muchas características de la base de datos de voz BIOMEX-DB están condicionadas por el protocolo empleado para la toma de señales de EEG.

4.2.1 DEMOGRAFÍA DE LOS VOLUNTARIOS

Las señales de voz fueron tomadas de 51 voluntarios de los cuales hay 26 hombres y 25 mujeres para asegurar una representación balanceada de ambos géneros para tener mayor variabilidad de voz.

Los voluntarios son personas que asisten al Instituto Nacional de Astrofísica Óptica y Electrónica (INAOE), entre trabajadores, estudiantes internos de posgrado y estudiantes visitantes.

La edad promedio de los voluntarios es de 29 años, esto se debe a que 35 de ellos tienen una edad que ronda entre los 20 y 30 años. Por otro lado, 16 años es la edad del voluntario más joven y 61 el de mayor edad. En [16] se menciona el hecho de que si el rango de edad es demasiado amplio entonces la tarea de verificación se facilita, sin embargo, ya se mencionó que hay un rango de edad consistente en un grupo mayoritario, por lo que no se facilita significativamente la verificación. En la figura 4.1 se muestra una gráfica que resume la información de edad de los hablantes.

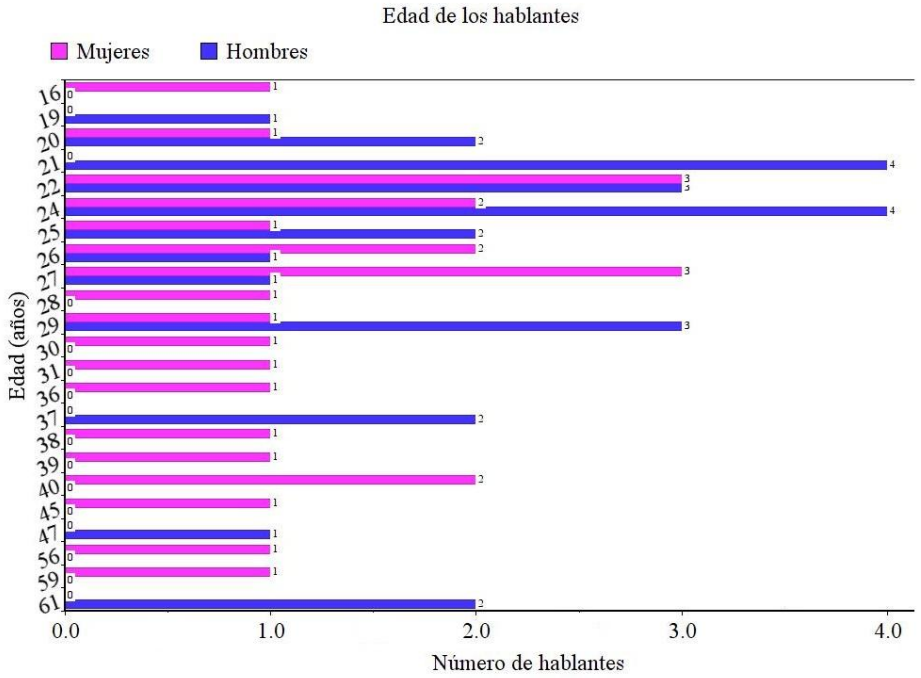


Fig. 4.1. Edades de los hablantes masculinos y femeninos de la base de datos BIOMEX-DB.

Del total de voluntarios, 44 son oriundos de México, de este grupo 23 son del estado de Puebla, mientras que el resto provienen de otros 14 estados de la república mexicana. Los 6 voluntarios extranjeros provienen de diversos países de Latinoamérica, a saber: Cuba, Colombia, Costa Rica, Ecuador y Venezuela.

4.2.2 CONTENIDO LÉXICO DE LA BASE DE DATOS DE VOZ

La base de datos de voz consiste en archivos de audio que contienen cadenas de dígitos comprendidos del 0 al 9, los dígitos se pronuncian de manera aislada y entre cada pronunciación de cada dígito hay un muy breve segmento de silencio como se observa en la figura 4.2.

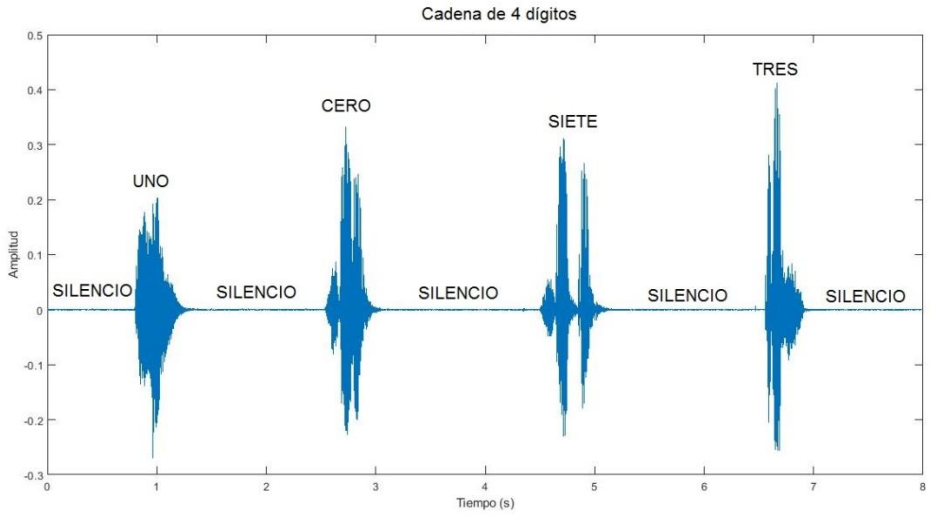


Fig. 4.2. Pronunciación de cadenas de 4 dígitos.

La base de datos de voz se divide en dos partes:

- La primer parte consiste en 10 cadenas diferentes de 10 dígitos cada uno, estas cadenas tienen el propósito de ser utilizadas para el entrenamiento de clasificadores. En cada cadena se pronuncia una sola vez cada dígito, lo que significa en esta parte hay 10 pronunciaciones de cada dígito por persona y 5100 dígitos pronunciados en esta parte de la base de datos.

- La segunda parte consiste en 10 cadenas diferentes de 4 dígitos, estas cadenas se utilizan para realizar pruebas al sistema de reconocimiento de hablante, a cada cadena se le considera una contraseña de 4 dígitos que es asignada a un hablante específico. En cada cadena se pronuncia una sola vez cada dígito, la distribución de repeticiones por persona para los dígitos cero, uno, dos, cuatro, cinco, seis, siete y nueve es de 4 repeticiones cada uno, mientras que hay 3 pronunciaciones del dígito ocho y 5 pronunciaciones del dígito tres.

  Los dígitos de todas las cadenas están ordenados aleatoriamente.

4.2.3 PROTOCOLO DE ADQUISICIÓN DE SEÑALES DE VOZ

La base de datos BIOMEX-DB fue grabada a lo largo de un mes y medio, todos los hablantes tuvieron una única sesión de grabación debido a limitantes de tiempo.

Todas las muestras de voz de cada hablante fueron tomadas de forma consecutiva en un espacio de tiempo de alrededor de 6 minutos, sin embargo, la sesión completa de cada hablante tuvo una duración aproximada de entre 15 a 20 minutos, debido a que se les colocaba y quitaba una diadema para tomar señales EEG.

Las grabaciones fueron hechas en un cuarto acondicionado para evitar la presencia de ruido externo, dentro del cuarto solo se encontraba un monitor con una cámara de video, un par de bocinas para mandar indicaciones auditivas, una silla para el voluntario y frente a esta se encuentra un micrófono marca Sennheiser modelo MD 421-II para captar su voz. Fuera del cuarto de grabación se encontraban equipos adicionales para adquirir las señales: el micrófono fue conectado a un amplificador Yamaha modelo MG06X, el amplificador a su vez fue conectado a la terminal de micrófono de una Workstation portátil marca Dell modelo M6700 con sistema operativo Windows 7 Service Pack 3 en la que se almacenaron los datos de voz.

Con un script de MATLAB se controlaron las rutinas para mostrar en pantalla varias secuencias de dígitos, grabar sonido desde el micrófono, generar los archivos de audio y las transcripciones correspondientes a cada archivo de audio.

Los voluntarios no tuvieron conocimiento previo sobre el protocolo de grabación hasta que iniciaba su sesión, al principio de esta se le colocó la diadema para EEG mientras se le daban las instrucciones que debía seguir durante su estancia en el cuarto de grabación:

1. Una vez colocada la diadema de EEG, el voluntario entra al cuarto de grabación para sentarse en una posición cómoda de forma que pueda atender a las indicaciones de la pantalla y ser captado por la cámara de video.

2. Cuando el voluntario dé la indicación de estar listo, la pantalla mostrará la instrucción de cerrar los ojos durante 15 segundos, después de ese lapso de tiempo sonará un timbre para que el voluntario abra los ojos y en pantalla se mostrará la instrucción de mantenerlos abiertos durante otros 15 segundos.

3. Terminado el paso 2 una cadena de dígitos aparecerán en pantalla, cada uno de ellos se mostrará por 2 segundos durante los cuales el voluntario debe pronunciar claramente el dígito que observa solo una vez. Se le hace énfasis al voluntario que su pronunciación sea natural.

4. Cuando una cadena de dígitos llegue a su fin aparecerá en pantalla durante 4 segundos la indicación de descansar, relajar su garganta y parpadear, el objetivo es hacer la sesión de grabación lo más cómoda posible. Pasados los 4 segundos se repite el paso 3.

5. Terminadas las 10 primeras cadenas de dígitos, se da un descanso de 1 minuto. Posteriormente se repite el proceso desde el paso 2 en adelante hasta pronunciar todas las cadenas de un segundo conjunto de cadenas de dígitos.

6. Terminada la sesión de grabación el voluntario sale del cuarto para quitarle la diadema y dar sus datos personales: la edad, su estado de procedencia si es mexicano o su país y provincia si es extranjero; y si es zurdo o diestro (para cuestiones de registro para las señales de EEG). Además de sus datos el voluntario puede dar alguna observación sobre su sesión.

Es importante aclarar que el paso 2 es parte importante del protocolo regular de adquisición de señales de EEG.

4.2.4 ESTRUCTURA DE LA BASE DE DATOS DE VOZ

La base de datos de voz está compuesta de 51 hablantes pronunciando 20 cadenas de dígitos cada uno para un total de 1020 pronunciaciones, un total de 7140 dígitos (140 por hablante).

En el apartado 4.2.2 se mencionó que existen dos conjuntos de cadenas de dígitos: un conjunto de 10 cadenas de 10 dígitos cada una y un conjunto de 10 cadenas de 4 dígitos cada una. El primer conjunto es de entrenamiento mientras que el segundo es para pruebas.

Los archivos son almacenados en una carpeta por hablante, el nombre de la carpeta inicia con la letra **F** o **M** para representar a un hablante femenino o masculino respectivamente, seguido de un identificador numérico que inicia con el dígito cero y el número de hablante, los identificadores tienen el valor de 1 al 25 en el caso de los hablantes femeninos y del 1 al 26 en el caso de los masculinos; por ejemplo, para el hablante femenino número 18 su carpeta tiene el nombre *F018*, mientras que para un masculino con el número 17 su carpeta tiene por nombre *M017*.

Los archivos que contienen información de un hablante en específico, sin importar su extensión se nombran de la siguiente manera:

- Primero se escribe la letra que identifica el género del hablante que pronunció la cadena junto con su identificador numérico como se hizo para nombrar carpetas, seguido de un guión bajo.

- Después del guión bajo se escribe con dos dígitos el número de sesión en el que se grabó el archivo de audio, ya que solo hubo una sesión todos los archivos tienen los números 01.

- Inmediatamente después del número de sesión se encuentra el identificador del grupo de cadenas al que pertenece el archivo, el valor *G10* indica que pertenece al grupo de cadenas de 10 dígitos y el valor *G04* significa que pertenece al grupo de cadenas de 4 dígitos. Después de este valor hay un guión bajo.

- Después del segundo guión bajo se escribe el número de la cadena que se pronunció, este identificador toma valores del 1 al 10 ya que cada grupo está conformado de 10 cadenas.

Cada cadena de dígitos tiene dos archivos asociados a ella, el primero corresponde a un archivo **.wav** que almacena la información de audio, mientras que el segundo archivo corresponde a las transcripciones de tiempo de las pronunciaciones de

la cadena cuya extensión es **.lab**. En total cada hablante tiene 20 archivos de audio y 20 archivos de transcripción.

Los archivos de audio fueron grabados con una frecuencia de muestreo de 16000 Hz y una resolución de 16 bits por muestra. Los archivos correspondientes al conjunto de cadenas de 10 dígitos tienen una duración cercana a los 20 segundos, mientras que los que pertenecen al conjunto de cadenas de 4 dígitos tienen una duración cercana a los 8 segundos, prácticamente todos los archivos que pertenecen a un mismo conjunto tienen la misma duración debido a la forma en que el script de MATLAB realiza el manejo de los datos de grabación de sonido. En total se tienen aproximadamente 238 minutos de grabaciones, aproximadamente 4.6 minutos por hablante.

Los archivos de transcripciones contienen los tiempos de inicio y terminación de una palabra o de un segmento de silencio, dichos tiempos tienen unidades de centenas de nanosegundo (x100ns), este formato es el utilizado por las herramientas de HTK.

A manera de ejemplo se presentan los nombres de los archivos de audio y transcripción de un hablante para una sola cadena:

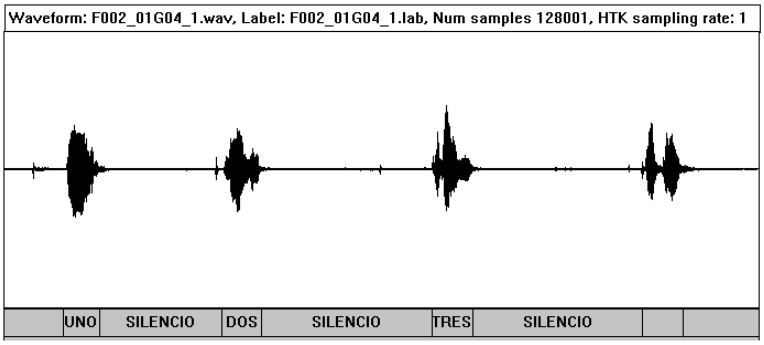
F018_01G10_3.wav

F018_01G10_3.lab

En la figura 4.3 a) se muestra la transcripción de un archivo de audio y en la figura 4.3 b) se observa un audio segmentado en dígitos de acuerdo a los tiempos del archivo de transcripción.

```
0 9226250 SILENCIO
9226250 12730000 UNO
12730000 24525625 SILENCIO
24525625 28379375 DOS
28379375 44846875 SILENCIO
44846875 48817500 TRES
48817500 65285000 SILENCIO
65285000 69138750 CUATRO
69138750 80000625 SILENCIO
```

a)



b)

Fig. 4.3. a) Contenido de un archivo de transcripción. b) Archivo de audio segmentado de acuerdo al archivo de transcripción.

En total la base de datos ocupa un total de 476 MB de espacio en un medio de almacenamiento.

Por último se menciona que también se tiene un archivo de registro en excel que contiene los datos de cada voluntario: su identificador numérico, género, edad, lugar de procedencia, si es zurdo o diestro, fecha y hora de su sesión; así como observaciones importantes de la sesión.

Fig. 4.4. Procedimiento de extracción de datos biométricos.

CAPÍTULO 5

ENTRENAMIENTO DE MODELOS, PRUEBAS Y EVALUACIÓN DEL SISTEMA DE VERIFICACIÓN DE HABLANTE

5.1 INTRODUCCIÓN

En capítulos anteriores se explicaron los conceptos teóricos de este trabajo y se definieron las características específicas del mismo.

En este capítulo se expondrán los detalles más importantes del entrenamiento de los HMM que representarán a los modelos globales y de los modelos de hablantes, la forma en que estos modelos serán utilizados para realizar la tarea de verificación de hablante dependiente del texto, también se profundizará en el procedimiento mediante el cual se evaluará el sistema de verificación y se presentarán los resultados arrojados por la evaluación.

5.2 RESUMEN DE LA TAREA DE VERIFICACIÓN DE HABLANTE

El presente trabajo de tesis tiene como objetivo realizar biometría de voz (definido en el capítulo 1 como reconocimiento de hablante), la tarea biométrica específica es la verificación de hablante dependiente del texto del tipo texto fijo.

Los modelos clasificadores elegidos para construir el sistema de verificación son los HMM de izquierda a derecha con densidades continuas de observaciones, los HMM fueron entrenados a nivel de palabra.

Los datos de entrenamiento son tomados de la base de datos **BIOMEX-DB** que fue creada internamente. Los vectores de características utilizados para entrenar los HMM son los MFCC junto con la primera y segunda derivada.

El entrenamiento es realizado con el método de reestimación de parámetros Baum-Welch y el algoritmo Viterbi.

También se implementará un sencillo esquema de comprobación de contraseña para añadir más seguridad al sistema.

La figura de mérito para evaluar el sistema es el LLR para obtener una calificación de la comparación de una muestra de prueba con un modelo entrenado del sistema de verificación.

Los resultados serán presentados en curvas ROC y en gráficas DET para mostrar un panorama general del desempeño del sistema.

5.3 ENTORNO DE TRABAJO

Para llevar a cabo las tareas de entrenamiento, pruebas y evaluación, es necesario crear un entorno de trabajo que permita hacer el manejo de archivos necesarios para dichas tareas, así como coordinar las herramientas de software disponibles.

El entorno de trabajo está constituido por varios módulos cuya función está definida y aceptan datos como argumentos de entrada y entrega un tipo específico de datos como salida. **El conjunto de módulos y de datos de entrada y salida constituyen el sistema de verificación de hablante**.

Los módulos están organizados en tres etapas: **el preprocesamiento**, **el entrenamiento de modelos** y **la evaluación de resultados**. Los módulos y el flujo de datos se muestran en la figura 5.1.

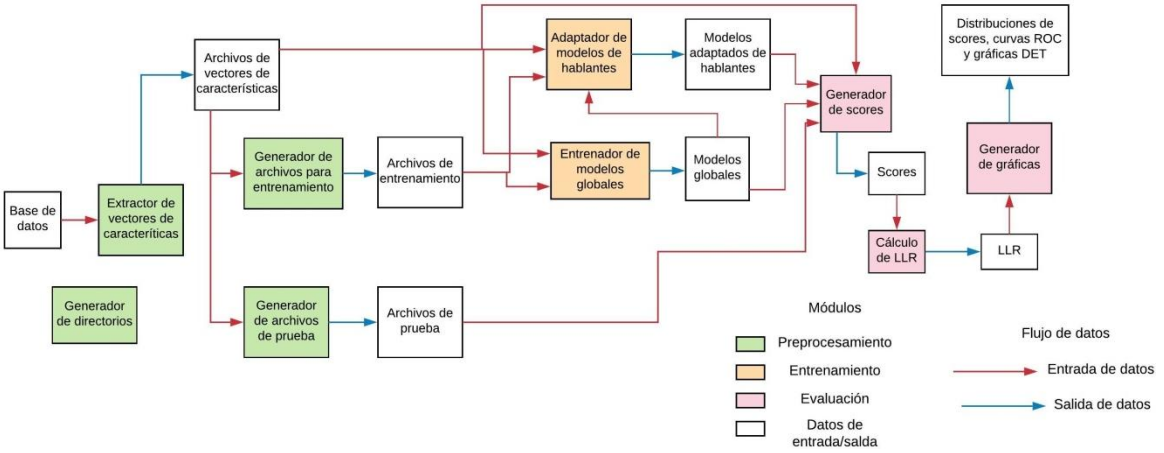


Fig. 5.1. Módulos del entorno de trabajo y flujo de datos.

Los módulos de la figura 5.1 están pensados cumplir las funciones de los módulos de un sistema biométrico presentados en el capítulo 1, con excepción del módulo de toma de decisión.

En los siguientes apartados se listarán las herramientas de software utilizadas y los scripts que se crearon para crear el entorno de trabajo.

5.3.1 EL CONJUNTO DE HERRAMIENTAS CYGWIN

En el apartado 3.4 del capítulo 3 se presentó el conjunto de herramientas HTK, al no estar integradas como una sola aplicación se deben compilar para adaptarse al sistema operativo y al hardware sobre el que se ejecutarán, además la ejecución de estas herramientas se hace mediante comandos a los que se les suministra argumentos.

Dado que todo el trabajo se realizó en el sistema operativo Windows 7, se pensó en realizar la compilación mediante la aplicación de símbolo de sistema, sin embargo,

esta opción resultó ser problemática por la cantidad de requerimientos de software para llevar a cabo la compilación.

La segunda opción es compilar en un equipo que tenga instalada una distribución de Linux, esta opción facilitaba en gran medida la compilación de HTK y otorgaba un entorno natural de línea de comandos para ejecutar las herramientas, la desventaja es que implicaba la instalación de este sistema y también el programa MATLAB para el manejo de resultados, por esta razón tampoco era viable.

La mejor solución se encontró en el conjunto de herramientas Cygwin, las cuales pueden ser instaladas en sistemas operativos Windows y proveen una funcionalidad similar a una distribución Linux en la forma de una línea de comandos y diferentes *shells*.

Se utilizó el *shell* de *bash* para realizar la compilación y ejecución de HTK, también se escribieron *scripts* para este shell para realizar diversas tareas de manejo de archivos y directorios.

5.3.2 *SCRIPTS* DE BASH PARA MANEJO DE ARCHIVOS Y DIRECTORIOS

Para realizar la creación y manejo del sistema de verificación de hablante se escribieron varios scripts para realizar de forma coordinada todas las actividades relacionadas con este propósito,

5.3.2.1 CREACIÓN DE DIRECTORIOS PARA ALMACENAR ARCHIVOS

El primer módulo, que contiene el *script directory_generator.bash*, crea todos los directorios necesarios para almacenar de manera organizada todos los archivos utilizados por el sistema.

La figura 5.2 muestra de manera general la jerarquía de directorios para organizar los archivos del sistema.

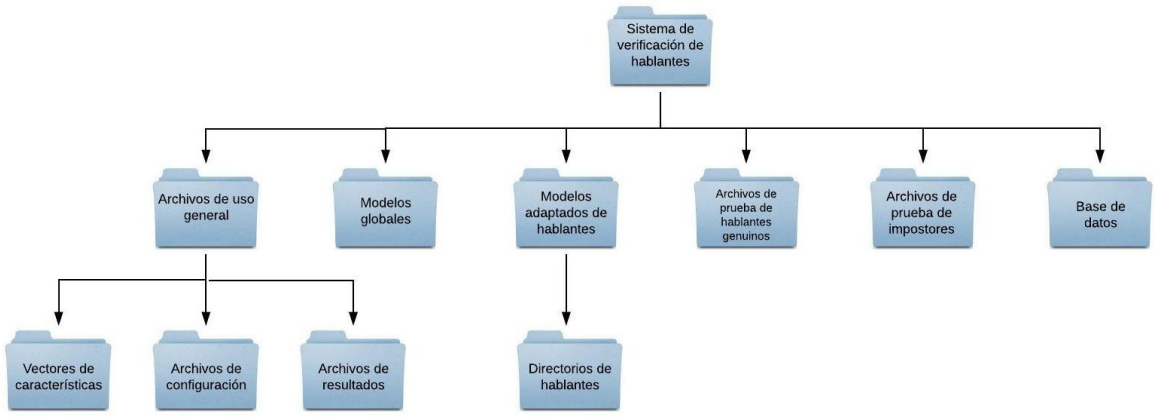


Fig. 5.2. Jerarquía de directorios para el sistema de verificación de hablante.

Esta organización resultó ser de suma importancia ya que todas las herramientas de software utilizadas requieren tener definidas las rutas de directorios para acceder a los archivos con los cuales llevan a cabo sus funciones.

5.3.2.2 MÓDULO EXTRACTOR DE CARACTERÍSTICAS

El segundo módulo corresponde al *script feature_generator.bash* que utiliza la herramienta *HCopy* de HTK, la cual puede extraer varios tipos de vectores de características basados en *cepstrum* utilizando los parámetros especificados en un archivo de configuración.

Para este trabajo se configuró la herramienta *HCopy* para extraer los MFCC de las señales de voz de la base de datos, los vectores de características obtenidos de un archivo de audio se almacenan en otro archivo con el mismo nombre del primero pero con extensión **mfc**. Más adelante se especificarán los parámetros de configuración utilizados para realizar la extracción.

5.3.2.3 GENERADORES DE ARCHIVOS PARA ENTRENAMIENTO Y PRUEBAS

Estos módulos están formados por varios scripts que generan archivos en un formato especial que las herramientas de HTK pueden utilizar, en el capítulo 3 de [36] se encuentra una muy breve descripción sobre los archivos que se utilizan para el proceso de entrenamiento y evaluación de modelos.

Estos módulos también organizan los directorios de los archivos con extensión mfc, de acuerdo a su función: entrenamiento y prueba; de tal manera que los módulos posteriores puedan utilizar esta información.

El generador de archivos de entrenamiento está formado por los scripts: *target_trainfiles.bash*, *UBM_trainfiles.bash*, *UBM_trainlabels.bash* y *prompts_UBM.bash*.

Por otro lado el generador de archivos de prueba tiene los siguientes scripts: *target_testfiles.bash* e *impostor_testfiles.bash*.

5.3.2.4 MÓDULOS DE ENTRENAMIENTO Y ADAPTACIÓN DE HMM

Como se mencionó en el capítulo 3, los modelos de hablantes se crean a partir de un proceso de adaptación, en el cual parámetros de los modelos globales independientes del hablante son reestimados utilizando los datos pertenecientes a un hablante particular.

Para estos propósitos se tienen dos módulos especializados para cada tarea. El módulo de entrenamiento de modelos globales está conformado por el *script training_ubm.bash*, mientras que el módulo de adaptación está formado por los *scripts training_adapted.bash* y *train_adapted.bash*; el primero realiza el proceso de adaptación mientras que el segundo coordina la selección tanto del modelo global como de los datos del hablante para la adaptación.

Cada módulo crea dos archivos, el primero almacena el HMM escrito en la sintaxis propia de HTK y el segundo almacena una matriz de varianzas asociada al HMM del primer archivo.

En un apartado posterior se detallará la forma en que se llevaron a cabo el entrenamiento y la adaptación.

5.3.2.5 GENERADOR DE *SCORES*(*), CÁLCULO DE LLR Y GENERADOR DE GRÁFICAS

Estos módulos constituyen la evaluación del sistema, permiten obtener cantidades numéricas que representan los resultados de pruebas al sistema e importar esas cantidades a otras herramientas de software para generar gráficas que permitan analizar el desempeño del sistema.

El generador de *scores* utiliza la herramienta *HVite* de HTK, la cual permite obtener el valor de $\log p(X|\lambda)$ resultante de poner a prueba un modelo global o de hablante con un determinado vector de características.

Los *scripts* que conforman este módulo son: *scores_target.bash*, *scores_impostors.bash* y *scores_generator.bash*; los dos primeros obtienen los valores de $\log p(X|\lambda_s)$ y $\log p(X|\lambda_{UBM})$ haciendo pruebas con hablantes genuinos e impostores respectivamente, mientras que el tercer script organiza los modelos que se utilizan para las pruebas así como los MFCC pertenecientes a hablantes genuinos e impostores.

Los dos primeros scripts también generan archivos de registro de cada prueba realizada al sistema, los cuales se pueden consultar para revisar si se realizaron correctamente, los *scores* son extraídos de los registros y almacenados en cuatro tipos de archivos, dos para scores de impostores que corresponden a pruebas con modelos de hablantes y modelos globales respectivamente; y dos para hablantes genuinos con las mismas condiciones.

El módulo de cálculo del LLR lo lleva a cabo de acuerdo a la ecuación (57), a diferencia de los módulos presentados anteriormente está conformado por scripts de MATLAB llamados *LLR_target.m* y *LLR_impostor.m* que importan los valores de scores de los archivos correspondientes y calculan el LLR, posteriormente crea dos archivos con los valores de LLR para hablantes genuinos e impostores.

La figura 5.3 ilustra claramente el funcionamiento de los dos módulos presentados.

Los módulos de generación de probabilidades logarítmicas y de cálculo de LLR son los que ejecutan la tarea de verificación de hablante, ya que son los que se encargan de calcular el *score* final (en nuestro caso el LLR), el cual se puede utilizar para hacer la comparación con un umbral determinado y tomar la decisión de aceptar o rechazar a un usuario del sistema.

Finalmente el módulo para generar gráficas se compone de scripts de MATLAB que importa los valores de LLR de los archivos de hablantes genuinos y de impostores para generar diferentes tipos de gráficas.

Cabe aclarar que la comparación con un umbral y la toma de decisión son acciones que dependen de las condiciones de la implementación real del sistema, por lo que su ejecución no está contemplada en ninguno de los módulos de evaluación; y como se mencionó en el apartado 5.2 el enfoque es tener un panorama general del desempeño del sistema.

(*)***Nota:*** en este trabajo a la calificación que se otorga a la comparación entre el vector de características y el modelo clasificador se le referirá con el término ***score***.

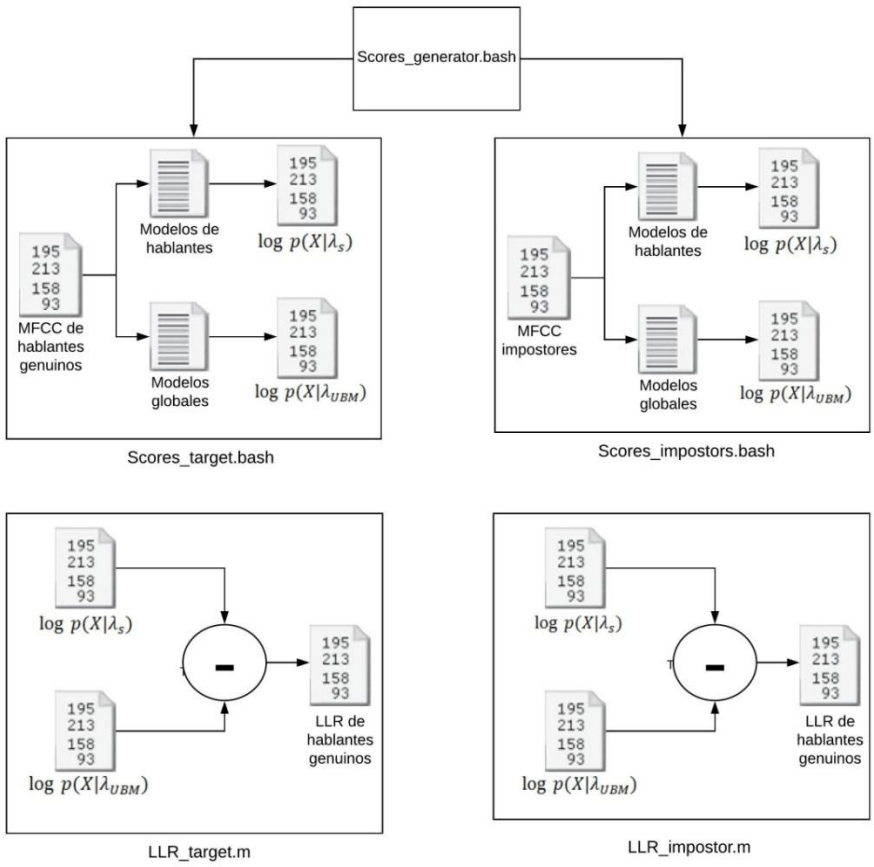


Fig. 5.3. a) Módulo generador de *scores*. b) Módulo de cálculo de LLR.

5.4 EXTRACCIÓN DE MFCC DE LAS SEÑALES DE VOZ

Para entrenar los HMM se utilizaron los vectores de características MFCC, en este apartado se presentan los parámetros de configuración para su extracción.

Con la herramienta *HCopy* de HTK, se obtuvieron los MFCC de todas las muestras de audio de todos los hablantes de la base de datos, con el fin de tener disponibles estos vectores para las etapas de entrenamiento y de prueba.

Por cada archivo de audio se genera un archivo de MFCC, los cuales tienen el mismo nombre que los archivos de audio pero con extensión .mfc como se mencionó

anteriormente, todos los archivos de MFCC son almacenados en el mismo directorio que los de audio y las transcripciones.

Para extraer los MFCC la herramienta *HCopy* requiere de un archivo de configuración en el cual se escriben los valores de los parámetros pertinentes, estos valores son similares a los utilizados en [6] y [23], la tabla 5.1 muestra los valores elegidos para los parámetros de extracción:

Tabla 5.1. Valores de los parámetros de extracción de MFCC.

Parámetro	Valor elegido
Tipo de fuente	Forma de onda
Formato de fuente	WAV
Tipo de vector de características	MFCC_C_0_D_A
Periodo de muestreo de la fuente	625
Tipo de ventana de segmentación	Hamming
Tamaño de la ventana (x100ns)	250,000
Traslape del ventaneo (x100ns)	100,000
Coeficiente de filtro pasa altas α	0.97
Número de filtros del banco	26
Número de MFCC	12
Normalización de energía	Sí

- El tipo de fuente especifica que es un archivo cuyos valores representan una señal de voz.
- Los vectores de características son MFCC junto con un coeficiente de energía y los coeficientes Delta y aceleración.
- La frecuencia de muestreo de los archivos de audio de entrada es de 16 KHz.
- El tamaño de la ventana para segmentar fue de 25 ms.
- El traslape del ventaneo es de 10 ms.
- Se especifica que los archivos de MFCC se almacenan en formato comprimido y se realice una comprobación de los mismos cuando se generan.
- El ancho de banda del banco de filtros se dejó con el valor predeterminado que es de 1 Hz a la frecuencia de Nyquist de la señal, es decir, 8 KHz.
- En [8] se muestra que no hay una mejora significativa del porcentaje de reconocimiento si se toman 12 o 26 MFCC.
- Se realiza una normalización de la energía de las señales de voz para compensar por la alta variabilidad de las señales.

El número de MFCC final fue de 13 (12 coeficientes del *cepstrum* y 1 coeficiente de energía), sumado a 13 coeficientes Delta y 13 coeficientes Delta-Delta. El tamaño total del vector de características fue de 39.

5.5 ENTRENAMIENTO Y ADAPTACIÓN DE HMM

La estructura de los HMM que se entrenaron para el sistema de verificación de hablante como ya se ha mencionado es la del modelo de izquierda-derecha con densidades continuas de observaciones, sin embargo, también ya se ha mencionado que HTK impone las condiciones de que el primer y último estado no generan ninguna observación; y un solo modelo de mezclas gaussianas por estado. Por esta razón, si se elige un modelo de 7 estados, en realidad se estará entrenando un modelo de 9 estados, pero los parámetros del estado 1 y del estado *N* son cero.

No hay una regla concreta para elegir el número de estados de un HMM para una tarea de reconocimiento, un factor importante es el nivel en el que se implementan los HMM (oración, palabra o fonema), una regla ampliamente difundida se encuentra en [7], la cual sugiere que el número de estados sea proporcional al número de fonemas que conforma el nivel de entrenamiento. Generalmente se eligen HMM de 3 estados para representar fonemas como en [7] o [36]; y hasta de 64 estados para una oración como en [7], en [4] se sugiere que para modelar a nivel de palabra se utilicen pocos estados.

Al igual que en el caso de número de estados, tampoco hay una regla escrita para el número de componentes que debe tener el modelo de mezclas gaussianas, si se utilizan demasiados componentes por mezcla no hay un método que pueda estimar cuántos de ellos contienen información de entrenamiento ni cuales componentes están entrenados, el problema que esto genera se refleja en costo de cómputo en caso de utilizar demasiados componentes.

Los HMM de este trabajo tienen 7 estados (9 para cumplir la condición de HTK). Debido a que la base de datos no es extensa se decidió utilizar solo 1 componente gaussiano en cada estado para facilitar la tarea de cómputo (en varios de los trabajos consultados el mínimo de componentes por estado es 16, sin embargo se debe considerar que las bases de datos que se utilizaron es esos casos consiste en varias decenas de hablantes con varias sesiones de enrolamiento), la matriz de varianzas del componente gaussiano es diagonal.

El entrenamiento se hace a nivel de palabra dadas las características de la base de datos. El objetivo es que cada HMM represente un dígito del 0 al 9, además también habrá uno que represente al silencio. Los datos de entrenamiento y adaptación fueron tomados de la primera parte de la base de datos **BIOMEX-DB**, que está conformada por las pronunciaciones de 10 cadenas de 10 dígitos cada una.

El objetivo es que cada hablante legítimo tenga un conjunto de 11 HMM que representen a su modelo de hablante y capturen su información acústica. Igualmente los modelos globales serán representados por 11 HMM entrenados con una gran cantidad de datos de muchos hablantes de la base de datos. La figura 5.4 muestra lo explicado anteriormente (los HMM de la figura solo son ilustrativos).

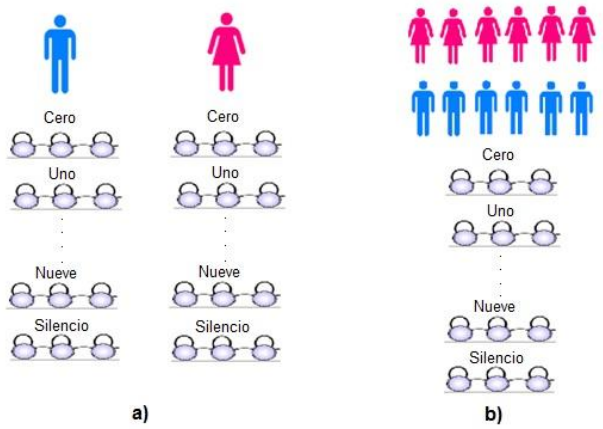


Fig. 5.4. a) Modelos de hablantes. b) Modelos globales.

5.5.1 ENTRENAMIENTO DE MODELOS GLOBALES

De acuerdo a [25] el modelo global representa la distribución de características independientes de los hablantes. Normalmente no es posible saber de antemano el género de un usuario (hablante genuino o impostor) que trate de usar el sistema de verificación, así que es recomendado utilizar **datos de entrenamiento independientes del género**.

En el mismo trabajo se discute la representación balanceada de ambos géneros dentro de los datos de entrenamiento para evitar una tendencia que favorezca a uno de ellos, también se discuten aproximaciones para organizar datos y de entrenar los modelos: la forma más básica es simplemente reunir todos los datos (balanceados por género) y usarlos para entrenar modelos globales independientes del género; otra alternativa es generar dos modelos dependientes del género y posteriormente unirlos para formar los modelos globales. Es posible aplicar estas aproximaciones a otros tipos de subpoblaciones de una base de datos, por ejemplo, subpoblaciones que representen los canales de grabación.

La primera aproximación fue la que se siguió en este trabajo debido a que no se cuenta con grandes cantidades de datos para entrenar modelos como se sugiere en la segunda aproximación. Dado que la base de datos **BIOMEX-DB** solo tiene un canal de grabación las únicas subpoblaciones consideradas fueron los géneros de los hablantes.

Otro problema inherente con estos modelos es determinar la cantidad adecuada de datos de entrenamiento, ya sea en cuestión de tiempo de los audios o en el número de hablantes, sin embargo, no hay una manera de exacta de determinar esta cantidad. Obviamente el tamaño de la base de datos que se use influye directamente sobre esta situación.

En este trabajo se hicieron distintas pruebas con diferente número de hablantes para entrenar los modelos globales con representación igualitaria de ambos géneros, esto se verá más adelante.

Para incorporar el aspecto de entrenamiento a nivel de palabra es necesario tener información etiquetada que permita localizar en el tiempo una palabra determinada en un

audio, es decir, contar con las transcripciones de tiempo. En [6] se entrenan dos conjuntos de modelos globales a nivel de palabra para verificación de hablante y se utilizan dos tipos de transcripciones para entrenar cada uno: generadas manualmente por personas y generadas por un sistema de reconocimiento de voz pre entrenado. Los resultados de verificación para ambos conjuntos no son muy diferentes entre sí, por lo que se puede asumir que ambas aproximaciones entregan buenos resultados. Sin embargo, en aplicaciones reales será difícil que se cuente con transcripciones generadas manualmente, por lo que la segunda aproximación es más viable en la mayoría de los casos.

Como se mencionó en el capítulo 4, todos los archivos de audio de la base de datos **BIOMEX-DB** cuentan con su transcripción correspondiente generada de manera automática con un script de MATLAB, por lo que se espera que no haya diferencia significativa a que se hubiera utilizado información generada manualmente.

Se definió un HMM prototipo creándolo con un *script* en '*Pearl'* incluido dentro de HTK, en este prototipo se definen el número de estados del HMM, el tipo y tamaño de los vectores de características con el que se va a entrenar, se inicializan los valores de las medias y las varianzas (el número de medias y varianzas es igual al tamaño del vector de características), se define el número de componentes gaussianas de cada estado; y finalmente se inicializan los valores de la matriz de transiciones.

El número de estados y los vectores de características ya fueron definidos con anterioridad, el vector de medias se inicializa con valores de 0 y las varianzas con 1, la matriz de transiciones se inicializa con valores de 0 para transiciones no existentes y con valores aleatorios en las transiciones existentes, con la condición de que los valores de cada fila sumen 1 excepto la última en la cual todos los valores deben ser 0. De forma predeterminada el número de componentes gaussianas por estado es 1 a menos que se definan más componentes, este valor no se modificó.

El entrenamiento se realiza en 7 iteraciones con distintas herramientas de HTK como sigue:

1. Se utilizó la herramienta *HCompV*, que calcula la media y covarianza de todos los datos de entrenamiento para los modelos globales y esos valores son asignados a todas las medias y covarianzas del HMM prototipo; el resultado es un nuevo HMM con la información global de los datos de entrenamiento.

2. La herramienta *HInit* utiliza el algoritmo Viterbi para encontrar la secuencia de estados más probable y estimar los parámetros del HMM.

   En *HInit* los datos de entrenamiento se segmentan uniformemente para todos los estados y se utiliza un el algoritmo de agrupamiento *K-means* para agrupar los datos en cada estado. Dado que solo se tiene una componente gaussiana, todos los datos asociados con un estado determinado se usan para estimar el peso del componente.

   Las transcripciones de tiempo permiten que la segmentación de datos se realice por palabra, por lo que el resultado de esta iteración son 11 HMM que representan los dígitos y al silencio.

3. La herramienta *HRest* toma los HMM inicializados por *HInit* y de forma individual reestima los parámetros de cada uno utilizando el algoritmo avance-retroceso y posteriormente el algoritmo Baum-Welch.

   Nuevamente las transcripciones permiten segmentar los datos de entrenamiento por palabra y el resultado son 11 HMM reestimados. Finalmente todos los HMM son concatenados en un solo archivo.

4. Se realiza un proceso de entrenamiento embebido en paralelo con la herramienta *HERest*, este entrenamiento reestima los parámetros de los HMM simultáneamente tomando el archivo que contiene todos los HMM.

   HERest utiliza las transcripciones para generar un HMM compuesto por los demás HMM ordenados de acuerdo a las transcripciones para formar la oración que se encuentra en ellas. Posteriormente se aplica el algoritmo avance-retroceso y Baum-Welch para la reestimación de parámetros, el resultado es un nuevo archivo que contiene todos los HMM reestimados.

5. Segunda reestimación con *HERest.*

6. Se utiliza la herramienta *HVite* para generar nuevas transcripciones mediante alineamiento forzado, que permiten a HTK tomar en cuenta varias pronunciaciones de una misma palabra o fonema para mejorar el reconocimiento. Ya que los dígitos de la base de datos solo tienen una pronunciación cada uno este paso podría omitirse, sin embargo, en [36] se recomienda llevarlo a cabo para mejorar los resultados de reconocimiento de voz, lo que puede significar también una mejora en la tarea de verificación de hablante.

   Con estas nuevas transcripciones se realiza una nueva reestimación con la herramienta *HERest.*

7. Utilizando las mismas transcripciones generadas en la iteración 6, se lleva a cabo la última reestimación de parámetros con *HERest*.

Al final del entrenamiento se tiene un archivo con todos los HMM de dígitos y de silencio con parámetros estimados, este archivo representa los modelos globales.

5.5.2 ADAPTACIÓN DE MODELOS DE HABLANTES

Como ya se explicó, para crear los modelos de hablantes los parámetros de los modelos globales se adaptan utilizando datos de entrenamiento de un hablante específico.

En [7] se muestra que el desempeño de un sistema de verificación se empobrece si se adaptan en conjunto las medias, los pesos de los componentes y las varianzas, en [7], [29], [30] y [37] solo se adaptan las medias para generar los modelos de hablantes.

Otro factor importante es el valor de τ, el coeficiente de ponderación de los datos de adaptación, en [7] se encuentra una tabla que muestra valores de τ dependiendo del

número de componentes gaussianas, siendo el valor de 100 el más común para sistemas con 16 componentes.

Para realizar la adaptación se utilizó nuevamente la herramienta *HERest* en una sola iteración, configurada para adaptar solamente las medias del componente gaussiano.

También se realizaron pruebas con distintos audios y distintos valores de τ para comparar los valores de $\log p(X|\lambda)$ resultantes, los mejores resultados de las pruebas se obtuvieron con τ en un rango de 10 a 15, el valor elegido fue de 12 ya que para valores mayores no hay mejora significativa mientras que valores muy altos empobrecen notablemente los resultados.

Los archivos de audio de la primera parte de la base de datos junto con sus transcripciones pertenecientes a hablantes genuinos fueron utilizados como datos de adaptación. Estos modelos son adaptados con 3 minutos y medio de datos de voz.

5.6 PRUEBAS AL SISTEMA DE VERIFICACIÓN DE HABLANTE

Una vez que se han entrenado el conjunto de modelos globales y de hablantes el siguiente paso es definir las condiciones bajo las cuales se llevarán a cabo las pruebas del sistema de verificación.

Debido a que no se cuenta con muchos datos de prueba es necesaria una forma de realizar pruebas que maximicen la cantidad de *scores* obtenidos como resultado para que representen el desempeño del sistema de verificación.

En [21] se exponen los detalles de un conjunto de pruebas conceptualmente similares a la **validación cruzada**, estas pruebas distribuyen los datos de entrenamiento y prueba dependiendo del propósito para el que se van a utilizar y realizan un proceso iterativo.

Se observó que las condiciones y detalles de las pruebas de [21] se podían adaptar a este trabajo y cumplir con el objetivo de obtener una cantidad suficiente de resultados para que sean estadísticamente significativos, así que se decidió que las pruebas para el sistema de verificación tengan los mismos criterios.

5.6.1 CARACTERÍSTICAS DE LAS PRUEBAS

Los *scores* resultantes de las pruebas provendrán de la comparación de los modelos del sistema con muestras de audio de los grupos de **hablantes legítimos** e **impostores**. Por esta razón se deben dividir los hablantes de la base de datos en estos dos grupos, además hay que considerar un tercer grupo cuyos datos servirán para entrenar los modelos globales.

Para llevar a cabo la división de hablantes en grupos, de datos de entrenamiento y prueba se tienen los siguientes lineamientos:

- Los hablantes fueron agrupados en 10 bloques, cada bloque tiene una cantidad igual y fija de hablantes.

- En cada bloque hay un determinado número de hablantes hombres y mujeres, esto dependerá de la cantidad total de hablantes que se tomen de la base de datos, más adelante se profundizará sobre esto.
- Los bloques a su vez se agrupan en tres categorías: **modelos globales** denotados como **UBM (6 bloques)**, **hablantes legítimos (2 bloques)** e **impostores (2 bloques)**.
- Un conjunto de pruebas tiene 20 iteraciones, en cada una cambian los bloques que conforman las categorías como se muestra en la tabla 5.2.
- Los archivos de cadenas de 10 dígitos de los hablantes que conforman las categorías de UBM y de hablantes legítimos, son utilizados para entrenamiento de los respectivos modelos.
- Los archivos de cadenas de 4 dígitos de los hablantes legítimos y de los impostores, son utilizados para poner a prueba los HMM del sistema.

Se observa que en cada iteración hay un cambio de bloques de las categorías de hablantes legítimos y de impostores, mientras que cada 2 iteraciones cambian los bloques de la categoría UBM, considerando lo anterior y el hecho de que los modelos de hablantes se derivan de los globales, se deben entrenar los primeros cada iteración; los segundos cada 2 iteraciones.

Las condiciones de estas pruebas permiten que todos los hablantes tomados de la base de datos hayan formado parte de alguna de las tres categorías al menos una vez y que los modelos de hablantes y globales se entrenen con distintos conjuntos de datos, con esto se asegura capturar toda la variabilidad acústica de los hablantes que conforman la población y que los resultados serán estadísticamente significativos.

Tabla 5.2. Agrupamiento de bloques de hablantes por iteración.

Iteración	Bloques UBM	Bloques Hablantes legítimos	Bloques impostores
1	1 – 6	7 – 8	9 – 10
2	1 – 6	9 – 10	7 – 8
3	2 – 7	8 – 9	10 – 1
4	2 – 7	10 – 1	8 – 9
5	3 – 8	9 – 10	1 – 2
6	3 – 8	1 – 2	9 – 10
7	4 – 9	10 – 1	2 – 3
8	4 – 9	2 – 3	10 – 1
9	5 – 10	1 – 2	3 – 4
10	5 – 10	3 – 4	1 – 2
11	6 – 1	2 – 3	4 – 5
12	6 – 1	4 – 5	2 – 3
13	7 – 2	3 – 4	5 – 6
14	7 – 2	5 – 6	3 – 4
15	8 – 3	4 – 5	6 – 7
16	8 – 3	6 – 7	4 – 5
17	9 – 4	5 – 6	7 – 8
18	9 – 4	7 – 8	5 – 6
19	10 – 5	6 – 7	8 – 9
20	10 – 5	8 – 9	6 – 7

Se realizaron 3 conjuntos de pruebas con distintos tamaños de población para observar el efecto que tienen sobre los resultados finales.

Los 3 conjuntos de pruebas tienen las siguientes condiciones:

1. La primera prueba se realiza con 30 hablantes, 15 hombres y 15 mujeres, los bloques son conformados por 3 hablantes cada uno, en cada bloque hay una combinación de 2 mujeres y 1 hombre o de 2 hombres y 1 mujer.

   Los modelos globales son entrenados con 1 hora de datos de voz pertenecientes a 18 hablantes.

   Hay 6 hablantes genuinos y 6 impostores.

2. La segunda prueba es con 40 hablantes, 20 hombres y 20 mujeres, los bloques tienen 4 hablantes cada uno, en cada bloque hay 2 hombres y 2 mujeres.

   Los modelos globales son entrenados con 1 hora 20 minutos de datos de voz pertenecientes a 24 hablantes.

   Hay 8 hablantes genuinos y 8 impostores.

3. La tercera prueba se realiza con 50 hablantes, 25 hombres y 25 mujeres, cada bloque tiene 5 hablantes, en cada bloque puede tener 3 hombres y 2 mujeres o 3 mujeres y 2 hombres.

   Los modelos globales son entrenados con 1 hora 40 minutos de datos de voz pertenecientes a 30 hablantes.

   Hay 10 hablantes genuinos y 10 impostores.

Los hablantes que conforman los bloques en cada conjunto de pruebas son elegidos al azar de la base de datos y cada población siempre está conformada por igual número de hombres y de mujeres.

En [19] y [30] se llevan a cabo tres tipos de prueba basadas en los tipos de errores de una tarea de reconocimiento. En este trabajo se realizaron tres tipos de pruebas: **hablante genuino equivocado** (conocido en inglés como *target wrong*, ocurre cuando un hablante genuino pronuncia una contraseña diferente a la que tiene asignada), **hablante genuino correcto** (en inglés *target correct*, ocurre cuando un hablante genuino pronuncia la contraseña que se le asignó) e **impostor correcto** (conocido en inglés como *impostor correct*, ocurre cuando un impostor pronuncia correctamente la contraseña de un hablante genuino).

Para implementar la prueba de hablante genuino equivocado primero se configuró la herramienta *HVite* para realizar un **alineamiento forzado**, esto consiste en generar la transcripción de un archivo de audio determinando el tiempo en el que se pronuncia una palabra específica, *HVite* permite agregar a la transcripción la probabilidad logarítmica de que una palabra determinada ocurra en un tiempo determinado, la figura 5.5 muestra un ejemplo de transcripción creada con alineamiento forzado.

```
0 10700000 SILENCIO -5321.479492
10700000 15100000 UNO -2831.896240
15100000 17100000 SILENCIO -1138.573853
17100000 21600000 DOS -2837.918457
21600000 25700000 SILENCIO -2097.330322
25700000 29600000 TRES -2421.289551
29600000 35500000 SILENCIO -3086.411377
35500000 39200000 CUATRO -2357.151123
39200000 39800000 SILENCIO -327.595306
```

Fig. 5.5. Transcripción creada mediante alineamiento forzado.

Para llevar a cabo el alineamiento forzado con *HVite* se requiere una **red de palabras**, esta red contiene las palabras que se quieren encontrar dentro de un archivo de audio en el orden en que se quieren encontrar. HTK tiene un formato definido para escribir un archivo que contenga la red.

Con el alineamiento forzado se buscan los dígitos de una contraseña dentro de un archivo de audio, si los dígitos pronunciados en el audio coinciden con los que se encuentran en la red y están en el mismo orden el valor de $\log\, p(X|\lambda)$ es bajo (indicando una probabilidad alta), en caso contrario el valor es alto. Dado que la base de datos tiene 10 cadenas de 4 dígitos se escribieron 10 redes de palabras con los dígitos contenidos en las cadenas. En la figura 5.6 se ejemplifica la red de palabras empleada para el alineamiento forzado, nótese que además de los dígitos se incluyeron los silencios, los dígitos se escribieron de acuerdo al contenido de las cadenas correspondientes de la base de datos.

Fig. 5.6. Red de palabras para alineamiento forzado.

Una vez configurado el alineamiento forzado la prueba de hablante genuino equivocado se realiza asignando una contraseña a cada uno de ellos, después cada hablante genuino pronuncia varias contraseñas equivocadas, los archivos de audio de las otras 9 cadenas de dígitos representan las contraseñas equivocadas, se calculan los LLR como se muestra en la figura 5.7.

En cada iteración se repite la prueba para cada una de las 10 contraseñas. Cuando se ejecutan las 20 iteraciones todos los valores de LLR se concentran en un mismo archivo llamado *targetpasswrong_LLR.txt.*

Cabe aclarar que para llevar a cabo la prueba de hablante genuino equivocado se hizo una modificación menor al *script* de *scores_target.bash*.

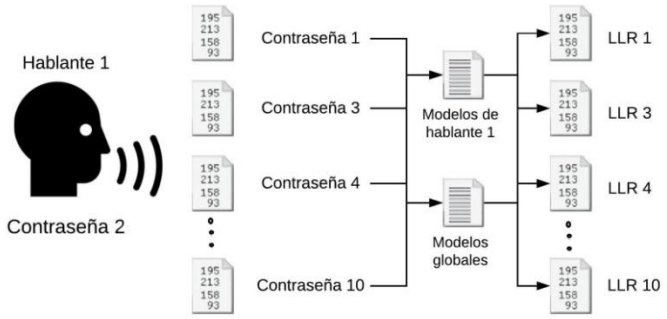


Fig. 5.7. Prueba de hablante genuino equivocado.

La prueba de hablante genuino correcto se realizó de manera similar a la de hablante genuino equivocado, en esta prueba cada hablante genuino pronuncia correctamente la contraseña que tiene asignada. Igualmente en cada iteración se repitió la prueba asignando las 10 contraseñas a cada hablante legítimo; los valores de LLR al finalizar las 20 iteraciones se guardaron en el archivo *target_LLR.txt.*

Finalmente para la prueba de impostor correcto también se hizo en condiciones similares a la primera, consiste en asignar una contraseña a cada hablante legítimo y los impostores tratan de ser aceptados por el sistema pronunciando correctamente la contraseña de cada hablante legítimo. Los LLR se almacenaron en el archivo *impostor_LLR.txt*. La prueba de impostores se hizo sin considerar el género, es decir, si un hablante genuino es mujer los impostores podían ser de ambos géneros, esto se justifica en la forma que se entrenaron los modelos globales.

La figura 5.8 ilustra el modo en que se hicieron las dos últimas pruebas.

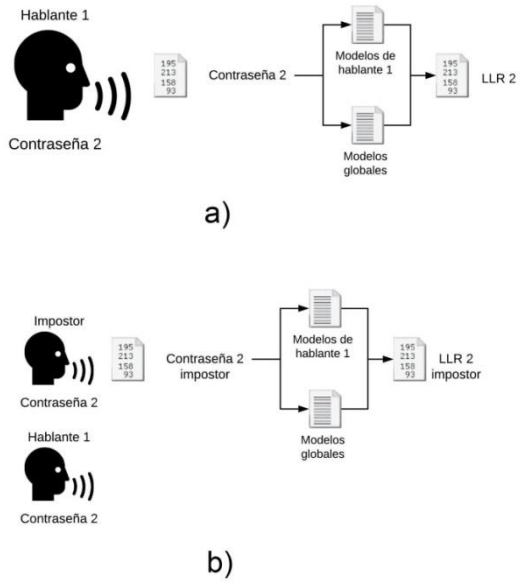


Fig. 5.8. a) Prueba de hablante genuino correcto. b) Prueba de impostor correcto.

Por último se presenta la cantidad de *scores* obtenidos de cada una de las pruebas y tomando en cuenta el tamaño de la población. Como se mencionó al principio del apartado 5.6 un objetivo muy importante de las pruebas es obtener un gran número de *scores* para que los resultados sean estadísticamente relevantes. Al observar las cantidades obtenidas se puede afirmar que este objetivo se cumplió satisfactoriamente, la tabla 5.3 muestra estas cantidades.

Tabla 5.3. Número de *scores* obtenidos por tipo de error y tamaño de la población.

Tipo de error / Población	Hablante genuino correcto	Impostor correcto	Hablante genuino equivocado
50 hablantes	2000	20,000	18,000
40 hablantes	1600	12,800	14,400
30 hablantes	1200	7200	10,800

5.7 EVALUACIÓN DEL SISTEMA DE VERIFICACIÓN

Para iniciar la parte de evaluación del sistema es importante considerar que no es posible hacer una comparación significativa de la precisión de varios sistemas de verificación dependiente del texto debido a que no hay un protocolo de evaluación estandarizado ni una base de datos estándar [16]. Por esta razón no se compararán los resultados obtenidos en este trabajo con aquellos que se encuentran en las referencias.

Al final del apartado 5.3.2.5 se aclaró que este trabajo no está enfocado en determinar un umbral de decisión ya que depende de las condiciones de una aplicación específica; se debe tener en cuenta que evaluando con un solo umbral solo se obtendrán aspectos cuantitativos del sistema de verificación tales como la cantidad de veces que se equivocó en clasificar a un hablante genuino o a un impostor dado un umbral.

La desventaja de solo tomar aspectos cuantitativos, es que no permite tener un panorama del comportamiento futuro del sistema. Por esto es necesaria una aproximación probabilística que describa el comportamiento del sistema con un rango de umbrales hipotéticos. Para hacer la evaluación del sistema de verificación desde un punto de vista probabilístico se siguieron las métricas presentadas en el apéndice 3.

Una vez definidas las métricas, se divide la evaluación en dos partes:

- La primera parte evalúa la precisión del sistema para aceptar hablantes genuinos y rechazar impostores, ambos tipos de usuarios pronuncian correctamente una contraseña previamente asignada a un hablante legítimo.

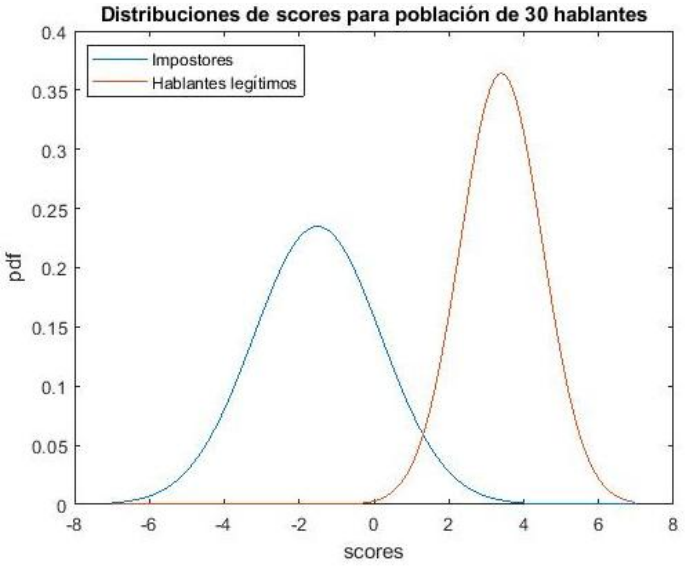
- La segunda parte evalúa la capacidad del sistema de verificar la identidad de un hablante legítimo mediante su contraseña, en este caso el objetivo es ratificar si el sistema rechaza a un hablante legítimo cuando no pronuncia su contraseña asignada.

En cada parte de la evaluación se tomará en cuenta los diferentes tamaños de población.

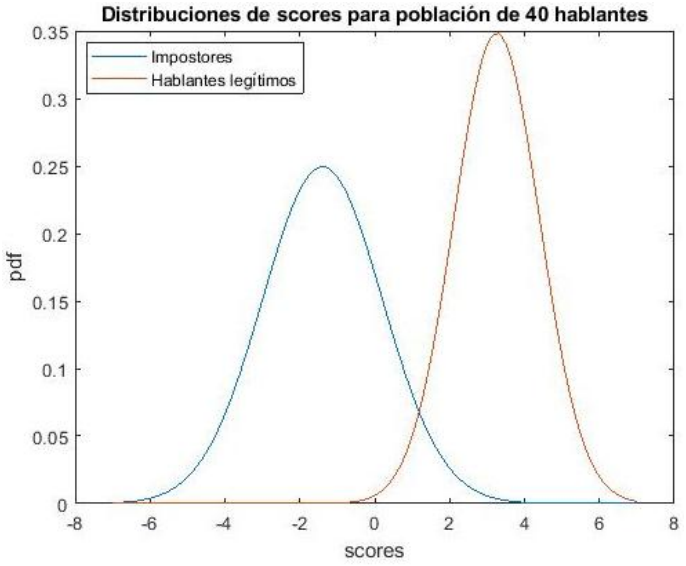
5.7.1 PRIMERA PARTE DE LA EVALUACIÓN

Para esta parte de la evaluación tomaremos los *scores* de hablante genuino correcto como la clase positiva y los de impostor correcto como la clase negativa. Dado que los *scores* de ambas clases fueron generados al pronunciar correctamente una contraseña asignada, la clasificación se basará en las características acústicas.

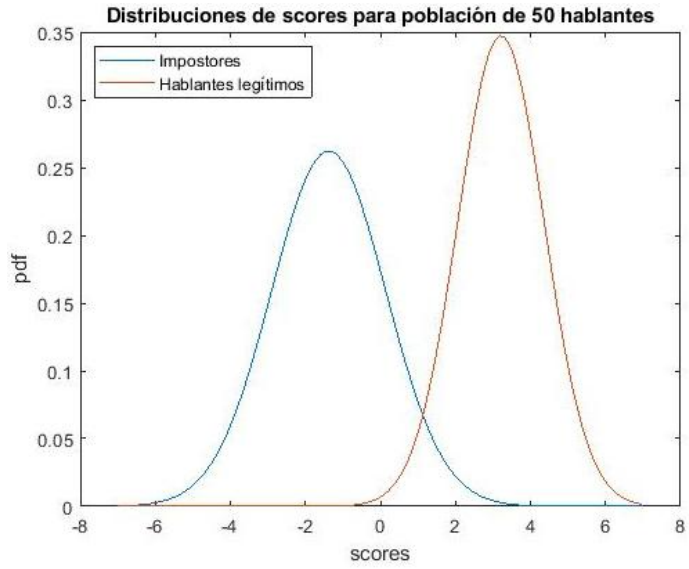
A continuación se presentan las distribuciones de *scores* de los distintos tamaños de población para observar la separabilidad entre las clases de hablantes genuinos e impostores.



a)



b)



c)

Fig. 5.9. Distribuciones de *scores* de desempeño del sistema de verificación para a) 30 hablantes. B) 40 hablantes. c) 50 hablantes.

Se puede observar buena separabilidad entre las dos clases para los tres tamaños de población, también se puede notar que el área correspondiente a los tipos de errores se incrementa ligeramente conforme aumenta el tamaño de la población.

A continuación se presentan las curvas DET de cada tamaño de población.

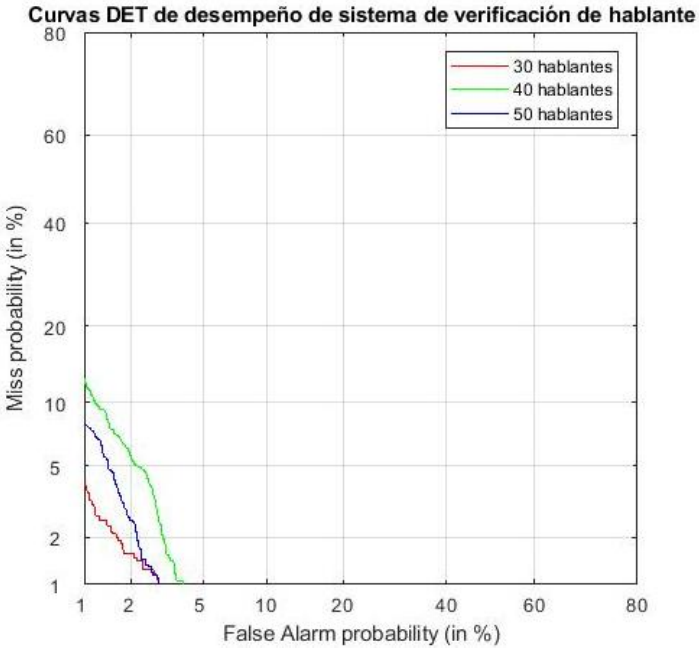


Fig. 5.10. Curvas DET de desempeño del sistema de verificación de hablante.

En estas curvas se observan un bajo porcentaje de errores, sin embargo, es evidente la tendencia de que el desempeño se empobrece conforme aumenta el tamaño de población.

Los valores de EER apoyan lo anterior, para la población de 30 fue de 1.75%, para 40 hablantes fue 2.81% y para 50 el valor fue de 2.13%.

Los valores de EER indican otro punto importante, a pesar del buen desempeño del sistema para cada tamaño de población, se observa que el EER de 40 hablantes es mayor que el de 50, lo cual contradice la noción de que el EER de 50 hablantes debería ser el de mayor valor.

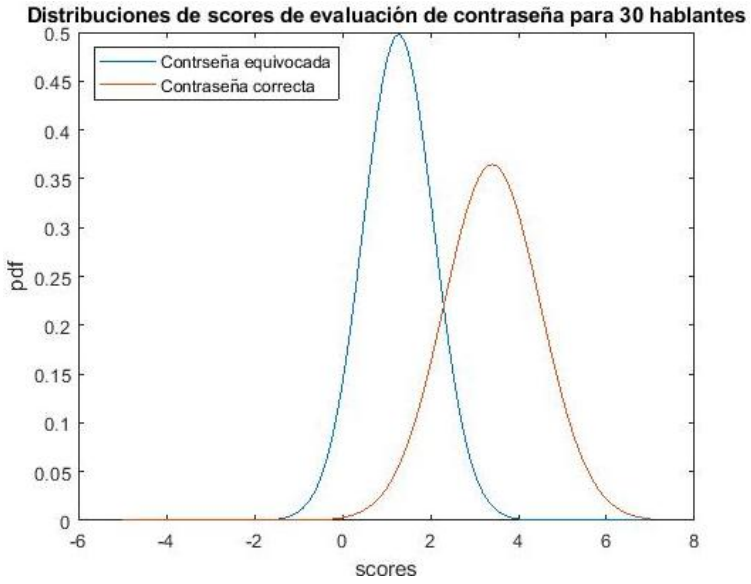
El buen desempeño del sistema de verificación se puede atribuir principalmente a que los audios de la base de datos **BIOMEX-DB** fueron grabados en un ambiente sin ruido, además de que el número de hablantes genuinos es bajo. Por otro lado, el hecho de que el sistema tenga menor desempeño para 40 hablantes que para 50 puede deberse a que la base de datos introduce un sesgo en los resultados.

Las curvas DET también permiten determinar un punto de operación del sistema de acuerdo al porcentaje que se pueda tolerar para ambos tipos de error.

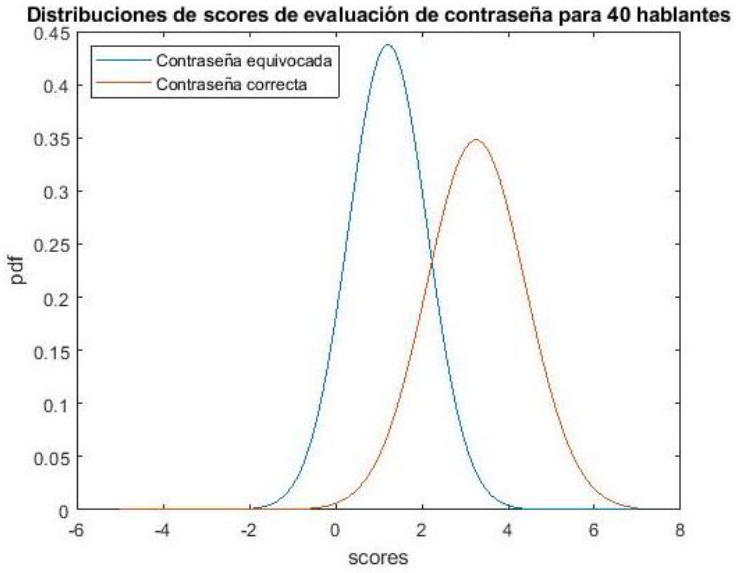
5.7.2 SEGUNDA PARTE DE LA EVALUACIÓN

La segunda parte tiene como propósito evaluar si el sistema puede distinguir si el hablante genuino pronuncia correctamente su contraseña asignada o no. En este caso la clasificación se basa en pronunciar correctamente la contraseña.

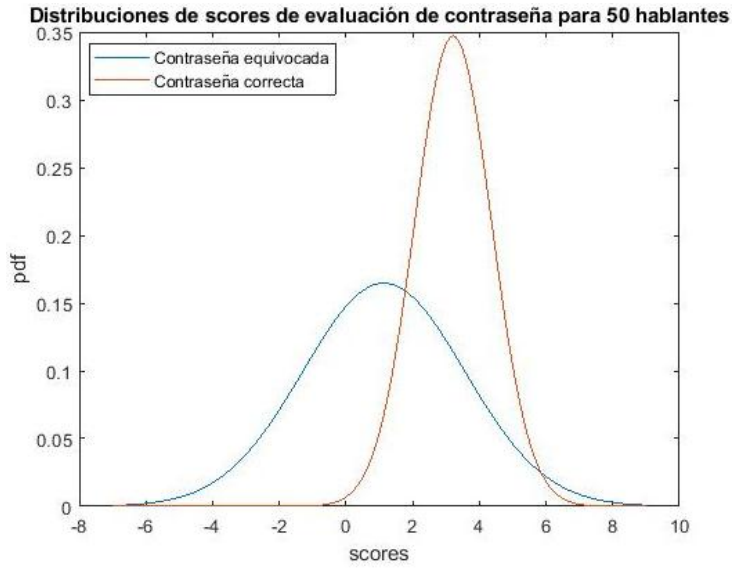
Los *scores* utilizados para esta parte son los de hablante genuino correcto como la clase positiva, mientras que los *scores* de hablante genuino equivocado son la clase negativa. Las distribuciones de *scores* se muestran a continuación.



a)



b)



c)

Fig. 5.11. Distribuciones de *scores* de evaluación de contraseña para a) 30 hablantes. b) 40 hablantes. c) 50 hablantes.

De las gráficas de las distribuciones se aprecian empalmes considerables, lo que indica mayor probabilidad de que el sistema acepte a un hablante genuino independientemente de que pronuncie correctamente su contraseña.

También se observa que hay un incremento de clasificación errónea conforme aumenta el tamaño de la población.

Para mostrar los resultados de la segunda de la evaluación se empleó la curva ROC, ya que sus parámetros (TPR y FPR) son más adecuados para esta evaluación.

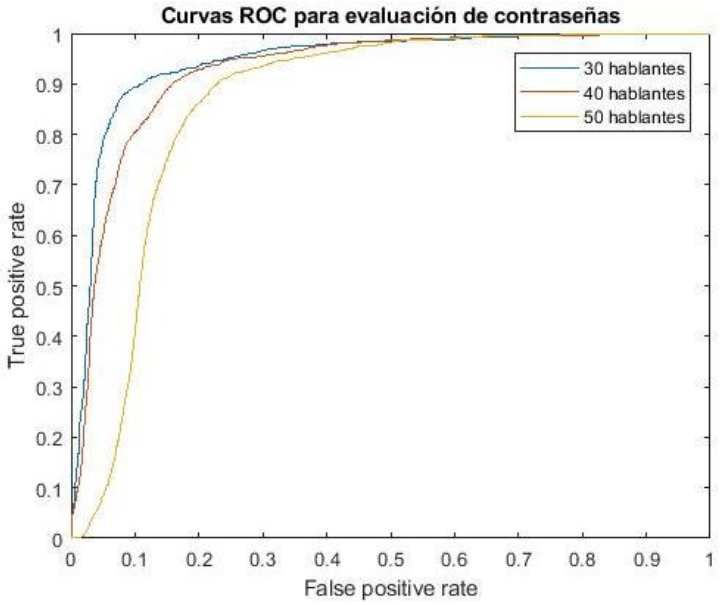


Fig. 5.12. Curvas ROC de evaluación de contraseñas para distintos tamaños de población.

Las curvas ROC muestran que el sistema de verificación tiene problemas para discriminar una contraseña correcta de una equivocada cuando son pronunciadas por un hablante genuino con dicha contraseña correcta asignada.

Estas curvas también corroboran el hecho de que conforme la población aumenta el desempeño se empobrece, aunque a diferencia de la parte de la evaluación que involucra hablantes genuinos e impostores, se observa claramente entre más población menor es el desempeño.

Finalmente se presentan los valores de EER de la segunda parte de la evaluación: 10.4% para población de 30 hablantes, 13.5% para 40 hablantes y 17.7% para 50.

5.8 CONCLUSIONES

En este trabajo se presentó un sistema de biometría por voz, cuya tarea principal es la verificación de la identidad de hablantes. Este sistema de verificación está basado en HMM para generar modelos globales que representan un conjunto de hablantes alternativos, los modelos de hablantes son derivados de los globales mediante una técnica de adaptación bayesiana. Los modelos globales fueron entrenados utilizando datos independientes del género.

Para el entrenamiento de los HMM se emplearon los MFCC como vectores de características, se extrajeron de los archivos de audio de la base de datos **BIOMEX-DB** que fue creada para este trabajo. Esta base de datos tiene un léxico compuesto por 10

dígitos comprendidos del 0 al 9. Los detalles y condiciones del entrenamiento fueron descritos exhaustivamente.

La tarea de verificación de hablante es dependiente del texto y para llevarla a cabo con la base de datos elegida se entrenaron los HMM a nivel de palabra, gracias a esta nivel de entrenamiento se implementó un conjunto de contraseñas de 4 dígitos para los hablantes genuinos para reforzar la seguridad.

Los *scores* empleados están basados en LLR obtenidos directamente de los modelos globales y de hablantes. Para obtener la mayor cantidad de *scores* se estableció un esquema de pruebas similar a la validación cruzada; el esquema de pruebas distribuyó la información de la base de datos en tres grupos de interés de forma iterativa para garantizar que los resultados sean estadísticamente relevantes.

Las pruebas que se llevaron a cabo están basadas en tres tipos de error: hablante genuino equivocado, hablante genuino correcto e impostor equivocado, estas pruebas fueron descritas detalladamente; las pruebas anteriores fueron hechas con diferentes tamaños de población.

Para la evaluación del sistema se emplearon un conjunto de métricas ampliamente utilizadas en sistemas de clasificación, para presentar la información de los resultados de las pruebas se utilizaron gráficas de distribuciones de *scores*, curvas ROC que es ampliamente utilizada en el campo de biometría y DET que es prácticamente un estándar para presentar resultados de verificación de hablante. Se empleó el EER como medida de desempeño general. La evaluación fue dividida en dos partes: la primer parte muestra el desempeño del sistema para clasificar a un usuario del sistema como un hablante genuino o como un impostor basándose en sus características acústicas; la segunda parte evaluó el comportamiento del sistema cuando un hablante pronuncia correcta o incorrectamente su contraseña.

Los resultados de la primera parte de la evaluación mostraron que el sistema de verificación tiene un buen desempeño para distinguir entre hablantes genuinos e impostores, este hecho se puede atribuir a las características de la base de datos, sin embargo, cuando el tamaño de la población de hablantes se incrementa este desempeño se degrada aunque no de manera drástica. Es importante mencionar que se halló evidencia de que la base de datos pudo haber introducido un sesgo en los resultados.

En el caso de la segunda parte se observó un comportamiento que indica que el sistema tiene dificultad para diferenciar una contraseña pronunciada correctamente de una incorrecta. Al igual que en la primer parte de la evaluación los resultados se iban degradando conforme aumentaba la población, la diferencia más notable es que este empobrecimiento fue consistente para los tres tamaños de población.

La evaluación de resultados presenta el panorama general del desempeño del sistema y entrega información sobre el rango de condiciones en las que puede trabajar si es que se decide implementarlo en una aplicación real.

Por último se hace mención que se crearon herramientas de software para llevar a cabo un control de las actividades del sistema de verificación: manejo de directorios y archivos, extracción de características, entrenamiento de modelos, ejecución de pruebas y evaluación de resultados. Estas herramientas consisten en *scripts* de bash y de MATLAB, dichos *scripts* emplean las herramientas de HTK y otros paquetes de software de terceros.

APÉNDICES

APÉNDICE 1 REGISTRO DE INFORMACIÓN DE LA BASE DE DATOS

ID. Hablante	Género (M/F)	Fecha (DD/MM/AA)	Hora (H:M)	Edad	Lugar de origen (Pais - Estado)	Zurdo o diestro
M001	M	17/07/2018	12:40 p.m.	61	México, Puebla	Diestro
M002	M	17/07/2018	12:57 p.m.	29	México, Puebla	Diestro
M003	M	17/07/2018	01:14 p.m.	47	México, Puebla	Diestro
M004	M	17/07/2018	01:30 p.m.	37	Ecuador, Quito	Zurdo
M005	M	17/07/2018	01:46 p.m.	27	México, Puebla	Diestro
M006	M	18/07/2018	12:16 p.m.	24	México, Puebla	Diestro
F001	F	18/07/2018	12:28 p.m.	22	México, Veracruz	Diestro
M007	M	18/07/2018	01:04 p.m.	37	Colombia, Bucaramanga	Diestro
F002	F	18/07/2018	01:22 p.m.	26	Mexico, Puebla	Diestra
M008	M	18/07/2018	01:41 p.m.	24	México, Tabasco	Diestro
M009	M	19/07/2018	11:39 a.m.	26	México, Sonora	Diestro
M010	M	19/07/2018	12:00 p.m.	25	México, Sonora	Diestro
F003	F	19/07/2018	12:17 p.m.	27	México, Puebla	Diestra
M011	M	20/07/2018	11:23 a.m.	22	México, Chihuahua	Diestro
F004	F	20/07/2018	11:38 a.m.	20	México, Sinaloa	Ambidiestra
M012	M	20/07/2018	11:54 a.m.	20	México, Guerrero	Diestro
M013	M	20/07/2018	12:09 p.m.	20	México, Sinaloa	Zurdo
M014	M	20/07/2018	12:26 p.m.	24	México, Puebla	Diestro
M015	M	20/07/2018	12:43 p.m.	19	México, Puebla	Ambidiestro
F005	F	20/07/2018	01:00 p.m.	24	México, Puebla	Diestra
F006	F	23/07/2018	10:51 a.m.	39	México, Puebla	Zurda
F007	F	23/07/2018	11:08 a.m.	40	México, Puebla	Diestra
F008	F	23/07/2018	11:26 a.m.	45	México, Puebla	Diestra
F009	F	23/07/2018	11:43 a.m.	38	México, Puebla	Diestra
M016	M	23/07/2018	12:02 p.m.	21	México, Sinaloa	Diestro
M017	M	23/07/2018	12:59 p.m.	21	México, Colima	Diestro
F010	F	23/07/2018	01:29 p.m.	22	México, Chiapas	Zurda
F011	F	24/07/2018	10:57 a.m.	28	México, Veracruz	Diestra
F012	F	24/07/2018	12:38 p.m.	25	México, Chihuahua	Ambidiestra
M018	M	24/07/2018	12:58 p.m.	24	Ecuador, Quito	Diestro
M019	M	24/07/2018	01:15 p.m.	21	México, Puebla	Diestro
M020	M	24/07/2018	01:38 p.m.	25	México, Puebla	Diestro
F013	F	25/07/2018	11:50 a.m.	56	México, Puebla	Zurda
M021	M	25/07/2018	12:36 p.m.	61	México, Puebla	Diestro
F014	F	25/07/2018	01:07 p.m.	40	México, Puebla	Diestra
M022	M	25/07/2018	01:28 p.m.	29	México, Puebla	Diestro
M023	M	26/07/2018	10:49 a.m.	22	México, Guanajuato	Diestro
M024	M	26/07/2018	12:12 p.m.	21	México, Sinaloa	Diestro
F015	F	26/07/2018	12:43 p.m.	24	Costa Rica, San José	Diestra
F016	F	26/07/2018	01:06 p.m.	22	México, Querétaro	Diestra
M025	M	26/07/2018	01:20 p.m.	22	México, Sinaloa	Diestro
F017	F	14/08/2018	11:13 a.m.	27	México, CDMX	Diestra
F018	F	14/08/2018	11:46 a.m.	16	México, Puebla	Diestra
F019	F	14/08/2018	12:21 p.m.	59	México, Puebla	Diestra
F020	F	14/08/2018	12:45 p.m.	31	Venezuela, Táchira	Diestra
F021	F	15/08/2018	10:43 a.m.	29	México, Oaxaca	Diestra
F022	F	15/08/2018	11:46 a.m.	26	Cuba, Camagüey	Diestra
F023	F	15/08/2018	12:26 p.m.	30	México, Sonora	Diestra
F024	F	15/08/2018	12:45 p.m.	27	México, Michoacan	Diestra
M026	M	15/08/2018	01:01 p.m.	29	México, Edo. Mex.	Diestro
F025	F	16/08/2018	12:25 p.m.	36	México, Puebla	Diestra

APÉNDICE 2 CONTENIDO LÉXICO DE LA BASE DE DATOS

Cadenas de 10 dígitos

1. SIETE NUEVE CERO DOS UNO CINCO OCHO SEIS CUATRO TRES
2. UNO SIETE CERO TRES OCHO CUATRO SEIS CINCO DOS NUEVE
3. SEIS OCHO DOS CINCO TRES CERO NUEVE UNO CUATRO SIETE
4. NUEVE CUATRO DOS UNO CERO TRES OCHO SIETE CINCO SEIS
5. DOS CERO NUEVE UNO TRES SIETE CINCO CUATRO SEIS OCHO
6. OCHO SEIS UNO CINCO SIETE CERO TRES NUEVE DOS CUATRO
7. TRES CINCO SEIS OCHO UNO DOS CUATRO SIETE NUEVE CERO
8. CUATRO TRES CINCO SEIS NUEVE SIETE CERO OCHO DOS UNO
9. CERO OCHO DOS UNO TRES NUEVE SIETE CUATRO SEIS CINCO
10. CINCO TRES UNO SEIS SIETE CERO CUATRO NUEVE OCHO DOS

Cadenas de 4 dígitos

1. UNO DOS TRES CUATRO
2. CINCO TRES DOS NUEVE
3. UNO CERO SIETE TRES
4. NUEVE SEIS CUATRO SIETE
5. CINCO CUATRO DOS UNO
6. OCHO TRES NUEVE SEIS
7. SIETE CERO SEIS OCHO
8. NUEVE CINCO DOS TRES
9. CERO SEIS CUATRO SIETE
10. OCHO UNO CINCO CERO

APÉNDICE 3 MÉTRICAS DEL DESEMPEÑO DEL SISTEMA DE VERIFICACIÓN DE HABLANTE

A3.1 DESEMPEÑO DE UN CLASIFICADOR

Verificación de hablante es un problema de clasificación con dos clases. Cada instancia de prueba es mapeada a solo un elemento del conjunto {***p***, ***n***} que son etiquetas de clase positiva y clase negativa. Un modelo clasificador mapea las instancias a una clase hipotética. Para referirse a las clases a las que la instancia pertenece hipotéticamente se usan las etiquetas {***Y***, ***N***} [38].

Dado un clasificador y una instancia hay cuatro posibles resultados [38]:

1. Si una instancia es de la clase positiva y es clasificada como positiva, el resultado es un **verdadero positivo** (*true positive*).

2. Si una instancia es de la clase positiva y es clasificada como negativa, el resultado es un **falso negativo** (*false negative*).

3. Si la instancia es de la clase negativa y es clasificada como negativa, el resultado es **verdadero negativo** (*true negative*).

4. Si la instancia es de la clase negativa y es clasificada como positiva, el resultado es **falso positivo** (*false positive*).

En la figura A3.1 muestra la matriz de confusión que se genera con las clases verdaderas, hipotéticas y los cuatro resultados antes mencionados.

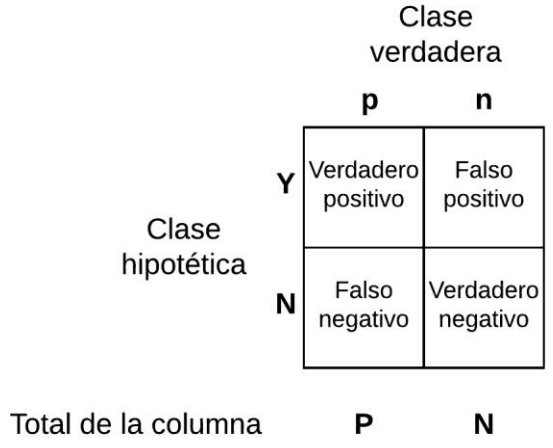


Fig. A3.1. Matriz de confusión de clasificación binaria [38].

De estos resultados se derivan varias métricas. Los resultados de la diagonal principal representan decisiones correctas, mientras que los otros dos resultados son errores entre las clases.

Las métricas derivadas son las siguientes [38]:

Tasa de verdaderos positivos (*true positive rate* ***TPR***).

$$TPR = \frac{Positivos\ clasificados\ correctamente}{Número\ total\ de\ positivos}$$

Tasa de falso positivo (*false positive rate* ***FPR***).

$$FPR = \frac{Negativos\ clasificados\ correctamente}{Número\ total\ de\ negativos}$$

Otros términos asociados:

$$Sensitividad = TPR$$

$$Especificidad = 1 - FPR$$

A3.2 EVALUACIÓN DEL SISTEMA DE VERIFICACIÓN DE HABLANTE

Para verificación de hablante existen dos condiciones para las pronunciaciones de entrada: ***s*** es la condición de que la pronunciación pertenece a un hablante genuino; y ***n*** que la pronunciación no pertenece a un hablante genuino. Igualmente dos condiciones existen: ***S*** es la condición de que la pronunciación es aceptada como perteneciente a un hablante genuino; ***N*** que la pronunciación es rechazada [10].

Las condiciones anteriores se pueden expresar en cuatro probabilidades condicionales que son análogas a lo descrito en el apartado A3.1 [10]:

1. *P(S|s)* es la probabilidad de aceptar a un hablante genuino, llamada **probabilidad de aceptación correcta** (*correct acceptance rate*).

2. *P(S|n)* es la probabilidad de aceptar a un impostor, llamada **probabilidad de aceptación falsa** (*false acceptance rate* ***FAR***).

3. *P(N|s)* es la probabilidad de rechazar por equivocación a un hablante genuino, llamada **probabilidad de falso rechazo** (*false rejection rate* ***FRR***).

4. *P(N|n)* la probabilidad de rechazar correctamente a un impostor, llamada **probabilidad de rechazo correcto** (*correct rejection rate*).

La tabla A3.1 es análoga a la matriz de confusión de la figura A3.1:

Tabla A3.1. Las cuatro probabilidades condicionales de verificación de hablante.

Condición de la pronunciación / Condición de decisión	Hablante genuino *s*	Impostor *n*
Aceptar *S*	*P(S|s)*	*P(S|n)*
Rechazar *N*	*P(N|s)*	*P(N|n)*

A3.2.1 DISTRIBUCIONES DE *SCORES*

La métrica del desempeño del sistema de verificación de hablante depende del número de pruebas que se realizan para la evaluación. Un número suficiente de pruebas es necesario para obtener una medida de evaluación estadísticamente significativa [11].

Las medidas de desempeño pueden estar basadas en *scores*, tomando en cuenta que en verificación de hablante es normal considerar dos clases: **hablantes genuinos** e **impostores**.

Las pruebas se realizan con instancias (pronunciaciones) de ambas clases y se obtienen *scores* correspondientes, posteriormente se lleva a cabo la clasificación comparando los *scores* con un umbral y se obtiene uno de los resultados presentados en el apartado A3.2. Con los *scores* se pueden generan un par de distribuciones de probabilidad. Estas distribuciones tendrán la tendencia a traslaparse cuanto más parecidas sean las dos clases, las áreas de traslape están asociadas a dos de los errores ya presentados [11]. En la figura A3.2 se muestra una gráfica con estas dos distribuciones.

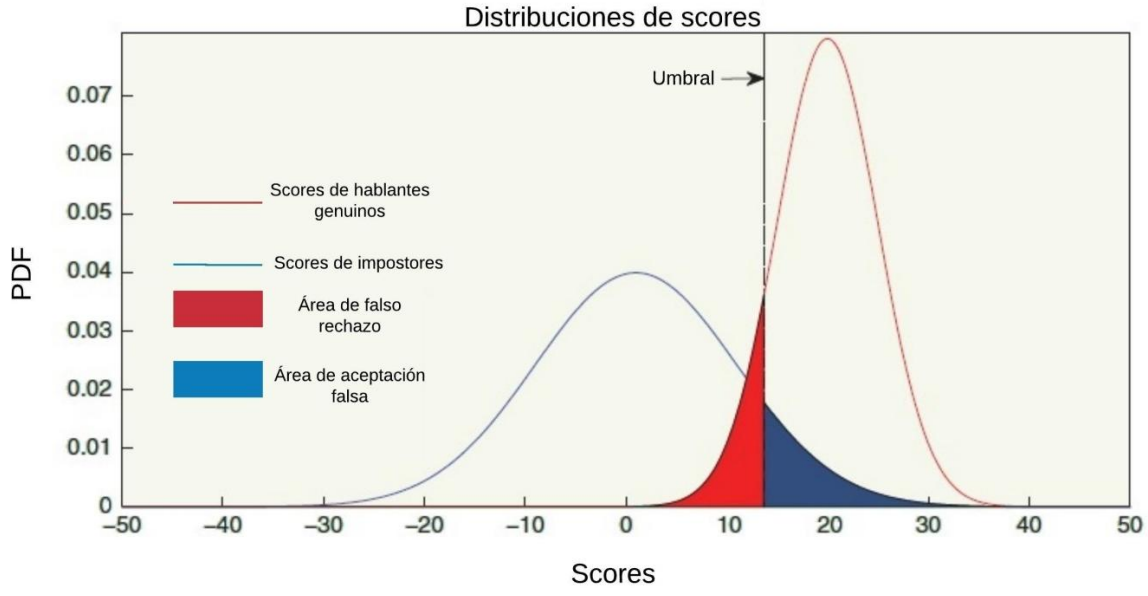


Fig. A3.2. Distribuciones de *scores* [11].

Entre más grandes sean las áreas de traslape, mayor será la probabilidad de que el sistema clasifique equivocadamente.

A3.2.2 LA CURVA ROC (RECEIVER OPERATING CHARACTERISTICS)

Las curvas ROC son gráficas de dos dimensiones en las cuales el eje Y representa el *TPR* y el eje X representa el *FPR.* Esta curva muestra el compromiso entre los beneficios (verdaderos positivos) y los costos (falsos positivos) [38]. Un ejemplo de curva ROC se muestra en la figura A3.3.

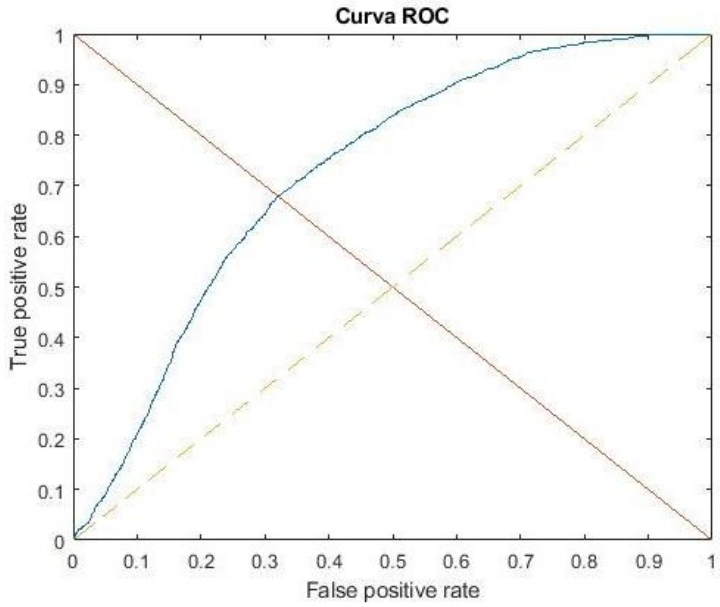


Fig. A3.3. Ejemplo **c**urva ROC.

Los clasificadores binarios producen el par *TPR* y *FPR* correspondiente a un punto en la curva ROC. Cada punto corresponde también a un solo valor del umbral para toma de decisión en verificación de hablante.

El punto (0,1) representa una clasificación perfecta, se considera que entre más cercano un punto a la esquina superior izquierda indica una mejor clasificación (el *TPR* es mayor al *FPR*), mientras que la cercanía a la recta comprendida por los puntos (0,0) y (1,1) (línea punteada en la figura A3.3) indica mayor aleatoriedad en la clasificación.

A3.2.3 LA CURVA DET (DETECTION ERROR TRADEOFF)

Esta curva es más utilizada cuando se requiere evaluar el desempeño del sistema de verificación de hablante en un rango de puntos de operación. Esta curva grafica en el eje X el *FAR* (identificados en esta curva como *False Alarm Probability*) contra el *FRR* que está en el eje Y (el *FRR* es identificado como *Miss Probability*).

En la figura A3.4 se muestra un ejemplo de curva DET.

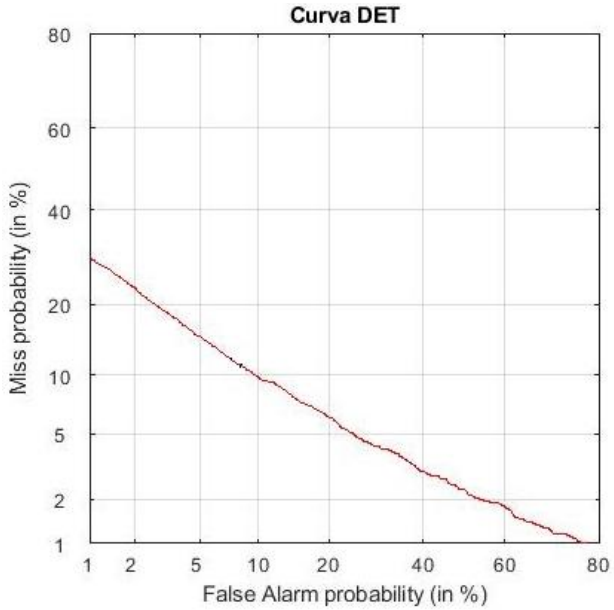


Fig. A3.4. Ejemplo de curva DET.

Mientras más cercana sea la curva al origen mejor es su desempeño.

Para generar esta curva se transforman las funciones de distribución acumulativa (FDA) de los *scores* de los hablantes genuinos y de los impostores en desviaciones normales. Esto significa que el valor de FDA de los *scores* de hablantes genuinos e impostores es transformado por una función de distribución acumulativa inversa y los valores resultantes se usan para generar la curva [11].

Para generar las curvas DET se utilizó la paquetería DETware_v2.1 para MATLAB (https://www.nist.gov/itl/iad/mig/tools).

A3.3 TASA DE ERROR IGUAL

La **tasa de error igual** (***Equal Error Rate*** en inglés, ***EER*** por sus siglas) es definido como los valores de *FAR* y *FRR* cuando ambos son iguales. Para igualar estos valores se cambia el valor del umbral de decisión hasta un punto en que se cumpla esta condición [11].

EL *EER* es una medida muy popular para evaluar el desempeño de un sistema de verificación de hablante. Aunque esta medida no indica un punto de operación óptimo, es ampliamente aceptado como medición de la capacidad de un sistema de verificación para distinguir hablantes genuinos de impostores.

En una curva ROC el punto donde se intersectan la curva misma y la recta diagonal (0,1) a (1,0) es el *EER*. En la curva DET es el punto donde se intersectan la curva y una recta de 45° que inicia en el origen.

REFERENCIAS

[1] Abdallah, S. J., Osman, I. M., & Mustafa, M. E. (2012). Text-independent speaker identification using hidden Markov model. *World of Computer Science and Information Technology Journal*, *2*(6), 203-208.

[2] Banerjee, A., Dubey, A., Menon, A., Nanda, S., & Nandi, G. C. (2018). Speaker Recognition using Deep Belief Networks. arXiv preprint arXiv:1805.08865.

[3] Beigi, H. (2011). *Fundamentals of speaker recognition*. Springer Science & Business Media.

[4] Benesty, J., Sondhi, M. M., & Huang, Y. (Eds.). (2007). *Springer handbook of speech processing*. springer.

[5] Bharti, R., & Bansal, P. (2015). Real time speaker recognition system using MFCC and vector quantization technique. *International Journal of Computer Applications*, *117*(1).

[6] Boakye, K. (2005). Speaker recognition in the text-independent domain using keyword hidden markov models. *Masters Report, University of California at Berkeley*.

[7] BÜYÜK, O., & Arslan, M. L. (2012). Model selection and score normalization for text-dependent single utterance speaker verification. *Turkish Journal of Electrical Engineering & Computer Sciences*, *20*(Sup. 2), 1277-1295.

[8] Debnath, S., Soni, B., Baruah, U., & Sah, D. K. (2015, January). Text-dependent speaker verification system: A review. In *Intelligent Systems and Control (ISCO), 2015 IEEE 9th International Conference on* (pp. 1-7). IEEE.

[9] Dunstone, T., & Yager, N. (2008). *Biometric system and data analysis: Design, evaluation, and data mining*. Springer Science & Business Media.

[10] Furui, S. (2007). Speech and speaker recognition evaluation. In *Evaluation of Text and Speech Systems* (pp. 1-27). Springer, Dordrecht.

[11] Hansen, J. H., & Hasan, T. (2015). Speaker recognition by machines and humans: A tutorial review. *IEEE Signal processing magazine*, *32*(6), 74-99.

[12] Ittichaichareon, C., Suksri, S., & Yingthawornsuk, T. (2012, July). Speech recognition using MFCC. In *International Conference on Computer Graphics, Simulation and Modeling (ICGSM'2012) July* (pp. 28-29).

[13] Kataria, A. N., Adhyaru, D. M., Sharma, A. K., & Zaveri, T. H. (2013, November). A survey of automated biometric authentication techniques. In *Engineering (NUiCONE), 2013 Nirma University International Conference on* (pp. 1-6). IEEE.

[14] Këpuska, V. Z., & Elharati, H. A. (2015). Robust speech recognition system using conventional and hybrid features of mfcc, lpcc, plp, rasta-plp and hidden markov model classifier in noisy conditions. *Journal of Computer and Communications*, *3*(06), 1.

[15] Khelifa, M. O., Elhadj, Y. M., Abdellah, Y., & Belkasmi, M. (2017). Constructing accurate and robust HMM/GMM models for an Arabic speech recognition system. *International Journal of Speech Technology*, *20*(4), 937-949

[16] Larcher, A., Lee, K. A., Ma, B., & Li, H. (2014). Text-dependent speaker verification: Classifiers, databases and RSR2015. *Speech Communication*, *60*, 56-77.

[17] Larcher, A., Lee, K. A., Ma, B., & Li, H. (2012). Rsr2015: Database for text-dependent speaker verification using multiple pass-phrases. In *Thirteenth Annual Conference of the International Speech Communication Association*.

[18] Laxmi Narayana, M., & Kopparapu, S. K. (2014). Choice of Mel Filter Bank in Computing MFCC of a Resampled Speech. *arXiv preprint arXiv:1410.6903*.

[19] Liu, Y., He, L., Tian, Y., Chen, Z., Liu, J., & Johnson, M. T. (2017, December). Comparison of multiple features and modeling methods for text-dependent speaker verification. *In Automatic Speech Recognition and Understanding Workshop* (ASRU), 2017 IEEE (pp. 629-636). IEEE.

[20] Martin, A., Doddington, G., Kamm, T., Ordowski, M., & Przybocki, M. (1997). *The DET curve in assessment of detection task performance*. National Inst of Standards and Technology Gaithersburg MD.

[21] Martın-Donas, J. M., López-Espejo, I., González-Lao, C. R., Gallardo-Jiménez, D., Gomez, A. M., Pérez-Córdoba, J. L., ... & Peinado, A. M. (2016). SecuVoice: A Spanish Speech Corpus for Secure Applications with Smartphones.

[22] Maurya, A., Kumar, D., & Agarwal, R. K. (2018). Speaker Recognition for Hindi Speech Signal using MFCC-GMM Approach. Procedia Computer Science, 125, 880-887.

[23] Ming, J., Hazen, T. J., Glass, J. R., & Reynolds, D. A. (2007). Robust speaker recognition in noisy conditions. *IEEE Transactions on Audio, Speech, and Language Processing*, *15*(5), 1711-1723.

[24] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257-286.

[25] Reynolds, D. A., Quatieri, T. F., & Dunn, R. B. (2000). Speaker verification using adapted Gaussian mixture models. *Digital signal processing*, *10*(1-3), 19-41.

[26] Ross, A. A., Nandakumar, K., & Jain, A. K. (2008). Handbook of biometrics. *US: Springer*.

[27] Sabhanayagam, T., Venkatesan, V. P., & Senthamaraikannan, K. (2018). A Comprehensive Survey on Various Biometric Systems. *International Journal of Applied Engineering Research*, *13*(5), 2276-2297.

[28] Saini, P., Kaur, P., & Dua, M. (2013). Hindi automatic speech recognition using htk. *International Journal of Engineering Trends and Technology (IJETT)*, *4*(6), 2223-2229.

[29] Sarkar, A. K., & Tan, Z. H. (2018). Incorporating pass-phrase dependent background models for text-dependent speaker verification. Computer Speech & Language, 47, 259-271.

[30] Sarkar, A. K., & Tan, Z. H. (2016). Text Dependent Speaker Verification Using Un-Supervised HMM-UBM and Temporal GMM-UBM. In *Interspeech* (pp. 425-429).

[31] Tiwari, V. (2010). MFCC and its applications in speaker recognition. *International journal on emerging technologies*, *1*(1), 19-22.

[32] Turk, U., & Schiel, F. (2003). Speaker verification based on the german veridat database. In *Eighth European Conference on Speech Communication and Technology*.

[33] Wang, J. C., Wang, C. Y., Chin, Y. H., Liu, Y. T., Chen, E. T., & Chang, P. C. (2017). Spectral-temporal receptive fields and MFCC balanced feature extraction for robust speaker recognition. Multimedia Tools and Applications, 76(3), 4055-4068.

[34] Wu, J. (2002). Hidden Markov Model. Seminar report of Machine Learning at Peking University, http://icl. pku. edu. cn/yujs.

[35] Youcef B.C., Elemine Y.M., Islam B., Farid B. (2017) Speech Recognition System Based on OLLO French Corpus by Using MFCCs. In: Chadli M., Bououden S., Zelinka I. (eds) Recent Advances in Electrical Engineering and Control Applications. ICEECA 2016. Lecture Notes in Electrical Engineering, vol 411. Springer, Cham.

[36] Young, S. (2009). The HTK book version 3.4. 1. *http://htk. eng. cam. ac. uk*.

[37] Barras, C., & Gauvain, J. L. (2003, April). Feature and score normalization for speaker verification of cellular data. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).* (Vol. 2, pp. II-49). IEEE.

[38] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, *27*(8), 861-874.

.