UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Reconocimiento del hablante en ambiente forense usando GMM's

TESIS

Que para obtener el título de

Ingeniero en Computación

P R E S E N T A

Héctor Adrián Zúñiga Sainos

DIRECTOR DE TESIS

Dr. Abel Herrera Camacho

Ciudad Universitaria, Cd. Mx., 2022

Agradecimientos

Al Dr. Abel Herrara Camacho por haberme dado la oportunidad de unirme al laboratorio de tecnologías de lenguaje y posteriormente siendo mi tutor.

A José Trangol por haberme proporcionado el software de reconocimiento utilizado durante los experimentos.

Al Consejo Nacional de Ciencias y Tecnología (CONACYT) por el apoyo a través del proyecto Caracterización de huellas textuales para análisis forense con clave 215179.

Índice

Introducción

El reconocimiento de hablantes ha ido creciendo durante el paso del tiempo, ya que las tecnologías de procesamiento de voz van mejorando con el paso de los años, sin embargo, al trabajar con la voz se pueden presentar dificultades que en otras medidas biométricas no. Por ejemplo, el estado de ánimo de una persona afecta directamente al tono que tiene su voz al hablar, gracias a esto el reconocimiento de hablantes puede tener problemas al efectuar el procedimiento.

Al tratarse de un ámbito forense, las condiciones pueden ser menos favorables por la falta de información para el entrenamiento del sistema, debido a que en este tipo de aplicaciones no es común contar con una grabación lo suficientemente larga o más de una para poder obtener el mayor número de características. Otro factor importante es la calidad del audio, si las grabaciones presentan una gran cantidad de ruido ambiental, el sistema se podría confundir cuando haga el reconocimiento, por esta razón es preferente tener un ambiente controlado para realizar las grabaciones y así aumentar el rendimiento del sistema, pero al igual que con la cantidad de información, las grabaciones presentadas no serán las más idóneas ya que se presentan en el momento más inesperado.

En esta tarea se utilizan métodos estadísticos para lograr discriminar de una mejor manera, ya que a diferencia de otros métodos que son completamente binarios, esto podría presentar mayor cantidad de resultados incorrectos porque al no poder generar un buen modelo por la falta de información, los centroides podrían no estar en el mejor punto y arrojar una cantidad mayor de errores. En cambio, con un modelo estadístico como en este caso, se crea una posibilidad para cada uno de los individuos en el corpus y lo que se pretende es encontrar el individuo que tenga la mayor probabilidad de que le pertenezca la voz.

A lo largo del tiempo se han utilizado diferentes métodos para la extracción de características de la voz, este trabajo se estará enfocando en el llamado Mel Frequency Cepstral Coefficient (MFCC) que ha demostrado brindar buenos resultados en la tarea. Por otro lado, para la parte de generación de los modelos de los individuos, existe un algoritmo llamado Gaussian Mixture Model (GMM) que creará dichos modelos con base en información estadística como la media y la covarianza. Por último, para la parte final que es la identificación, se utilizará una técnica llamada Maximum Likelihood Estimation (MLE) que dirá cuál es la identificación final. Dicho método será utilizado para comprobar que tan confiable puede llegar a ser en el reconocimiento del hablante, utilizando un corpus llamado Valquiria.

Durante esta tesis se revisarán métodos que se han utilizado desde los inicios del reconocimiento de hablantes; la forma de trabajar una señal de voz para poder ser procesada y posteriormente se

pueda realizar el reconocimiento por medio de las técnicas de extracción de coeficientes MFCC, creación de los modelos con GMM y la evaluación con MLE. Finalmente se explicará el programa utilizado para realizar estas tareas, los experimentos y los resultados obtenidos.

Capítulo 1. Antecedentes

A través de los años las personas hemos sido capaces de poder distinguir rasgos característicos de otras, uno de los más importantes es la voz ya que ha permitido a la humanidad comunicarse desde hace siglos. Logramos diferenciar la voz entre una y otra para así determinar qué persona es la que se dirige a nosotros, logrando entablar una conversación con la misma.

Uno de los grandes sueños que se busca alcanzar mediante el uso de sistemas computacionales, es el ser capaces de desarrollar un software que actúe de la misma forma en que lo hacemos nosotros con nuestras habilidades naturales para la comunicación. Es por eso que durante los últimos años se ha llevado a cabo una gran investigación en el área de procesamiento de voz, aunque se está muy lejos de lograr dicha meta.

A pesar de que aún no somos capaces de crear un sistema que logre procesar lenguaje natural, se han realizado muy buenas aproximaciones mediante diferentes técnicas de procesamiento de señales, modelos estadísticos y reconocimiento de patrones. En 1952 vió la luz uno de los primeros sistemas de reconocimiento de voz, cuando Davis, Biddulph y Balasheck crearon en los laboratorios Bell el sistema “Audrey”, que permitía reconocer dígitos de forma aislada para un solo hablante[1].

Un área de investigación en el procesamiento de voz, es el reconocimiento automático de hablantes y éste se puede dividir en dos tareas básicas. Una de ellas es la identificación de hablantes, se pretende con una entrada de voz poder reconocer si el hablante se encuentra dentro de una base de datos, de ser así, el sistema debe decirnos quién es la persona o en el caso contrario debe informar que no se encuentra; la segunda es la verificación de hablantes, aquí lo que se pretende es confirmar la identidad de la persona que está hablando, teniendo un modelo que contiene características de la voz de dicho sujeto, se realiza una comparación para determinar si la persona es quién dice ser o si se trata de un impostor.

Ambas tareas se pueden realizar utilizando dos métodos. El método de texto independiente [24] consiste en que, sin importar lo que el hablante diga al sistema, se puede hacer el reconocimiento de manera adecuada. Entre las características que se extraen se encuentran la auto-correlación, matriz de covarianza, histogramas de frecuencia fundamental, coeficientes de predicción lineal, entre otros.

Los métodos dependientes de texto se diferencian de los independientes debido a que, para realizar el reconocimiento, el hablante debe de decir una serie de palabras o una frase específica para poder realizar la comparación y determinación de la decisión; estos al ser limitados en la entrada, se logra obtener mejores resultados que en los métodos independientes de texto.

1.1 Décadas 60's y 70's

Fue en la década de 1960 cuando se empezaron a realizar los primeros sistemas de reconocimiento automático de hablantes[1]. En los laboratorios Bell, Pruzansky fue uno de los primeros en realizar dicha investigación utilizando bancos de filtros y correlacionando dos espectrogramas digitales. Posteriormente, Pruzansky y Mathews fueron capaces de mejorar el sistema haciendo uso de discriminantes lineales.

Atal, en el año de 1974, creó un sistema basado en las características de predicción lineal de las señales de audio [2]; para la extracción se utilizó una frecuencia de muestreo de 10 kHz. La finalidad de la predicción lineal es el de tratar de modelar la resonancia que se produce en el tracto vocal en el momento que nosotros hablamos.

Para lograr determinar los coeficientes del predictor se asume que ni el tracto vocal, ni la señal tienen cambios considerables de una muestra a otra, por lo tanto, se calculan minimizando el error de predicción.

Los datos utilizados en este sistema consistían de 60 instancias, cada una con seis repeticiones de la misma frase hablada por 10 mujeres diferentes. Las grabaciones se realizaron en 2 días para posteriormente volverlas a hacer en 27 días. Ya con las grabaciones hechas, las señales pasaron por un filtro paso bajas con un rango de 10000 Hz y posteriormente cuantizadas en números binarios de 12 bits.

Cada instancia fue dividida en 40 segmentos, la duración de cada uno fue hecha proporcional a la duración de cada instancia. El análisis de predicción lineal produjo un predictor de 12 coeficientes para cada uno de los 40 segmentos de todas las instancias.

En la determinación de qué tan parecido es un vector de prueba con uno de referencia, se introdujo una medida de distancia entre los vectores. Para la tarea de identificación se realiza el cálculo con cada hablante de la población y el vector de entrada es asociado a aquel cuya distancia sea la mínima. En la verificación se hace un proceso diferente, la distancia entre el vector de entrada y el supuesto hablante se compara con un umbral, si el valor de distancia obtenido es menor que el umbral, el hablante es aceptado, de lo contrario se rechaza.

Los resultados obtenidos para el experimento de identificación tuvieron una exactitud del 63.8 %; mientras que para la verificación se consideró a un hablante como el que se iba a verificar y a los otros nueve como impostores, para una duración de 0.2 segundos 90 %, para 0.5 segundos 95 % y para 1 segundo 98%.

Doddington [3], en Texas Instruments desarrolló un sistema de verificación de hablantes automático. Haciendo uso del método dependiente de texto, los usuarios tenían que hablar cuatro palabras seleccionadas de manera aleatoria, dentro de un conjunto de dieciséis palabras monosilábicas. Las señales de entrada al sistema son pasadas por un banco de filtros y posteriormente las salidas son muestreadas cien veces por segundo y digitalizadas. Cada salida de los filtros es cuantizada a tres bits.

Para cada palabra de prueba se especifican puntos clave, tales son elegidos en las regiones donde se encuentra la máxima energía en cada porción de las vocales para cada palabra. Para tener un

patrón de referencia, se toman seis muestras de amplitud espectral cada 20 ms en un intervalo de 100 ms. De esta forma cada frase es asociada a cuatro patrones de referencia.

Para la etapa de verificación, cada uno de los patrones de referencia se escanea a lo largo de la amplitud de los vectores muestreados en la frase de entrada y se localizan los puntos clave correspondientes en las frases de prueba. Durante este escaneo se realiza un cálculo de error cada 10 ms entre los datos espectrales de entrada y sus correspondientes patrones de referencia, con esto se obtiene una función de error para cada patrón de referencia que permitirá determinar la localización de cada punto clave en la entrada.

Una vez que se tienen identificados todos los puntos clave la función de decisión es únicamente el promedio de error durante el escaneo y un estimado de error esperado para el hablante en cuestión. El error estimado es calculado durante una etapa inicial de entrenamiento. Si los puntos clave no se pueden localizar, se procede a pedir otra frase y realizar el mismo proceso.

1.2 Décadas 80's

En el año de 1985 Soong hizo uso de la cuantización vectorial en el reconocimiento de hablantes. En el artículo publicado [4] explica que se puede hacer uso de un vector para representar las características acústicas, fonológicas o fisiológicas de un hablante si durante el entrenamiento se incluyen las suficientes variaciones.

Para obtener dichos vectores usó los vectores LPC en tiempo corto y para realizar el cálculo de la distancia entre ellos, agregó en la ecuación una matriz de correlación del vector de entrada, con el vector asociado.

En este proyecto creó un codebook, para lograr esto se debe de particionar el vector de entrenamiento de cada hablante; posteriormente cada una de dichas particiones se representaron por un vector que sirvió como centroide.

El proceso desarrollado consiste en obtener la señal de entrada de cada hablante y muestrearla para así llevar a cabo el análisis LPC. Con los vectores LPC obtenidos se realizó una cuantización utilizando N codebooks correspondientes a cada uno de los N hablantes. Se calcula un error de cuantización con respecto a cada codebook. Finalmente se comparan con la entrada para encontrar la menor distancia y así encontrar al hablante.

La base de datos utilizada en su experimento consistía de 100 hablantes, de cada uno de ellos se grabó 200 dígitos aislados. Para los experimentos se dividieron en dos. El primero usando 1, 2, 4 y 10 dígitos diferentes. El otro tomando en cuenta el tamaño del codebook, se tomaron 6 con 2, 4, 8, 16, 32 y 64 vectores.

Los mejores resultados obtenidos fueron cuando se utilizaron 10 dígitos y un codebook de 64 vectores; el rango de reconocimiento alcanzado con estas características fue de un 98 %.

Los laboratorios Bell crearon un sistema enfocado al trabajo con líneas telefónicas que fuera capaz de tolerar la variabilidad y degradación producida por las condiciones ambientales, así como las que se presentan durante la transmisión.

El funcionamiento general del sistema empieza con una señal de entrada que es digitalizada a una frecuencia de muestreo de 10 kHz. Para realizar el análisis de intensidad la entrada se pasa por un filtro paso-bajas a 900 Hz. Para el predictor de coeficientes o análisis de formantes la entrada es sujeta a un filtro paso-bajas a 3000 o 4000 Hz. Con los análisis realizados, se obtiene una curva de la muestra de entrada que posteriormente se comparará con las curvas de referencia asociadas al hablante.

Para determinar si se acepta o rechaza al hablante, se realizó un cálculo de distancias entre la curva de la muestra y las curvas de referencia, una vez obtenida, esta distancia se compara con un umbral para saber el resultado. Dicho umbral se estima con la ayuda de los conjuntos de un individuo y un conjunto de muestras de un impostor.

Durante los años de 1980 se buscó la posibilidad de hacer el uso de los modelos ocultos de Markov (HMM) con los métodos dependientes de texto. HMM se había utilizado previamente en el procesamiento de voz, pero posteriormente se encontró que poseía las mismas ventajas para el reconocimiento de hablantes.

En los métodos independientes de texto también se hizo uso de HMM, aunque esta vez se combinó con la cuantización vectorial. Lo que aportaba el uso de la cuantización vectorial es que se podían crear puntos representativos de las características de la voz.

Poritz [25] propuso caracterizar una entrada como una secuencia de transiciones mediante cinco estados en un espacio de características acústicas. Más tarde, Tishby decidió aumentar esta idea usando ocho estados representados por funciones de densidad de probabilidad continua con dos a ocho componentes mixtas por estado, lo cual tenía una mayor resolución espectral comparado con el método utilizado por Poritz. Rose propuso usar únicamente un estado y a eso es a lo que ahora se le conoce como Modelo Mixto Gaussiano (GMM).

1.3 Década 90's

Durante el año de 1992 Younès Bennani [5] propuso un sistema híbrido para combinar perceptrones multi-capa y los modelos ocultos de Markov. Se hizo uso de la base de datos TIMIT para la realización del experimento.

Durante la etapa de entrenamiento, los perceptrones nos brindan en sus capas ocultas una codificación de las sentencias, las cuales son una representación de la señal de voz de entrada y que a su vez contienen información discriminante del hablante. Esta codificación se puede utilizar posteriormente como entrada para crear los modelos de los correspondientes hablantes.

El primer módulo del sistema está encargado de brindar un modelado de los hablantes, después los modelos pasan por una etapa de clustering de acuerdo a la distancia Itakura. Esto sólo es para la inicialización del algoritmo ya que, posteriormente, se sustituye por una red neuronal y que el reconocimiento se haga de una manera más rápida. La red se entrena con cada uno de los clusters para identificar a los hablantes de la población. Por último, se manda a la última capa oculta donde se codifica y se obtiene una representación comprimida de la señal de entrada; dicha salida será la entrada a un HMM.

Los resultados obtenidos con este sistema híbrido fueron bastante buenos. Con 10 mujeres se obtuvo un reconocimiento de 99.6%, con 10 hombres fue de 99.6%, con 15 mujeres se logró un 99.7%, pero su mejor resultado se alcanzó con 22 hombres y un porcentaje de identificación del 99.8%.

En 1995 Reynolds [6] utiliza GMM para implementar un sistema de identificación y verificación. En la realización del sistema primero segmentó en cuadros por ventanas cada 20 ms. Utilizó un detector de actividad de voz para eliminar los cuadros donde hubiera ruido o silencio, ya que para aplicaciones independientes de texto es de suma importancia deshacerse de dichos cuadros, tanto en el entrenamiento como en las pruebas, para poder modelar y detectar únicamente el hablante.

Una vez que se eliminaron los cuadros no deseados, se obtienen los vectores con las características cepstral en la escala mel de los cuadros de voz. Por último, los vectores son ecualizados mediante una deconvolución ciega. Esta se obtiene promediando el vector cepstral de cada una de las palabras.

En el caso de identificación se usa un clasificador de máxima similitud. Dado un conjunto de hablantes S, cada uno con su respectivo modelo λ, se busca encontrar la máxima similitud con el vector de características, X, obtenido de la señal de entrada. La regla de decisión está dada por una suma de logaritmos y usando el teorema de Bayes.

$$\hat{s} = \arg \max_{1 \le s \le S} \sum_{t=1}^{T} \log p(x_t|\lambda_s) \qquad \ldots . (1)$$

Para la verificación se tiene que realizar un rango de similitud para determinar si la identidad del hablante se acepta o no. Se debe tomar la probabilidad de que sea el hablante en cuestión dado una entrada y compararlo con la probabilidad de que no sea el hablante dada la misma. El resultado obtenido de la operación es comparado con un umbral y si lo supera el hablante se acepta, en caso contrario se rechaza.

Autor	Características	Método	Texto
Atal, 1974	Predicción Lineal (LP)	Comparación de patrones	Dependiente
Markel y Davis, 1979	LP	Términos estadísticos largos	Independiente
Furui, 1981	Cepstrum normalizado	Comparación de patrones	Dependiente
Schwartz, 1982	Log Area Ratio (LAR)	Función de densidad de probabilidad (PDF) no paramétrica	Independiente
Li y Wrench, 1983	LP, Cepstrum	Comparación de patrones	Independiente
Doddington, 1985	Banco de filtro	Alineamiento temporal dinámico (DTW)	Dependiente
Soong, 1985	LP	Cuantización Vectorial (VQ)	10 dígitos aislados
Higgins y Wohlford, 1986	Cepstrum	DTW	Independiente
Attili, 1988	Cepstrum, LP, autocorrelación	Términos estadísticos largos	Dependiente
Higgins, 1991	LAR, LP-cepstrum	DTW	Dependiente
Tishby, 1991	LP	HMM	10 dígitos aislados
Reynolds, 1995	Mel-Cepstrum	GMM	Dependiente
Che y Lin, 1995	Cepstrum	HMM	Dependiente
Colombi, 1996	Cepstrum	HMM	Dependiente
Reynolds, 1996	Mel-Cepstrum	GMM	Independiente
Reynolds, 2000	GMM-UBM	GMM-UBM	
Campbell, 2006	GMM-SVM	GMM-SVM	
Kenny, 2004	JFA	JFA	
Dehak, 2009	i-vector system	i-vector system	

Tabla 1.1 Métodos de reconocimiento de hablantes.

En 1996 Colombi [7] realizó un sistema utilizando modelos ocultos de Markov. La base de datos utilizada consistía en una serie de frase, cada una con 3 pares de números que van del 21 al 99. Así, se podía incorporar una gramática al modelo de leguaje utilizando el algoritmo de Viterbi. En la extracción de características se filtraron ventanas de 25 ms, analizadas cada 10 ms, usando 24 filtros en la escala-Mel, después se aplicó una transformación coseno para representar 12 coeficientes MFCC.

En la identificación de hablantes se utiliza un clasificador bayesiano, asumiendo prioridades iguales, se escoge un modelo de hablante i de una instancia. Cada hablante es representado por 21 modelos monofonema de tres estados, dando como resultado 4914 parámetros por cada hablante. Dado que esa cantidad de parámetros era muy grande se realizó una reducción de características.

Durante la verificación, como en trabajos previos, la toma de decisión se realiza con base en el teorema de Bayes. Para esto se utiliza la diferencia entre las probabilidades de que sea el hablante en cuestión y la probabilidad de que se trate de un impostor, una vez obtenido dicho resultado, se compara con un umbral que puede ser único para todos los hablantes o se puede tener uno por cada uno.

Después de la realización del experimento se llegó a resultados muy buenos. Primero para la identificación tuvo un error de tan solo 0.22% entre una población de 138 hablantes, mientras que en la verificación, con la misma población, tuvo un error de 0.28%.

Con el paso de los años se han desarrollado diferentes sistemas de verificación e identificación, extrayendo diferentes características y utilizando varios métodos; han obtenido buenos resultados al cumplir con sus respectivas tareas. En la tabla 1.1 [8][9] se muestra de manera cronológica algunos de estos sistemas.

1.4 Década 2000's

Reynolds en el año 2000 [10] desarrolló un sistema de verificación de hablantes haciendo uso de una variación de los modelos mixtos Gaussianos (GMM), dicha variación se le conoce como Gaussian Mixture Model-Universal Background Model (GMM-UBM), así mismo, utilizó una técnica de normalización llamada HNORM que mejora el desempeño del sistema cuando se entrena y se realizan las pruebas con diferentes micrófonos.

La tarea de verificación se basa en determinar las relaciones de similitud de la probabilidad de que la entrada pertenezca a un hablante S sobre la probabilidad de que la entrada no pertenezca al hablante S, una vez computado este valor es comparado con un valor de umbral θ, si se supera o iguala dicho umbral se acepta que la entrada corresponde a ese hablante, de lo contrario se rechaza.

El tratamiento que se le dio a los datos antes de ser utilizados en el GMM-UBM es el siguiente: primero se realizó una segmentación en ventanas de 20 ms. Posteriormente se pasó por un detector de actividad de voz para poder descartar las partes que contienen ruido o silencio, este detector se basa en la energía de la señal para realizar la detección de las partes no correspondientes a la voz y descartando entre un 20% y 25% del total. Después se extraen los vectores de características cepstral en escala-mel. La escala-mel cepstrum es la transformada de coseno discreta de las energías log-spectral del segmento de la voz.

Para GMM se utilizan matrices diagonales de covarianza, en lugar de matrices completas porque pueden alcanzar el mismo nivel de desempeño y son más eficientes computacionalmente. Con los vectores de entrenamiento se calculan los parámetros del modelo de máxima similitud utilizando el algoritmo de máxima esperanza.

Uno de los métodos más utilizados en los últimos años es Support Vector Machine (SVM) y en el año de 2006, Campbell desarrolló un sistema de verificación independiente de texto con esa técnica [11]. SVM es un clasificador de dos clases construido a partir de las sumas de una función kernel K. La salida deseada es 1 y -1 que corresponden a la clase 1 ó a la clase 2 respectivamente. De igual manera que otros métodos la toma de decisión se obtiene mediante la comparación del valor obtenido de la función kernel con un umbral.

Para los experimentos utiliza el corpus NIST del 2005, la extracción de características es un vector de dimensión 19 de coeficientes MFCC determinados a partir de una señal de voz con filtro pre-énfasis cada 10 ms, con ventanas de Hamming de 20 ms. Se aplicó un detector de voz basado en energía para descartar los vectores con energía baja.

Para la función kernel se usa un supervector GMM el cual se obtiene del mapeo de una instancia con un vector de mayor dimensión. Para poder implementar el kernel y que cumpliera con la condición de Mercer, en lugar de usar directamente la divergencia, se utiliza una aproximación, dando como resultado un kernel lineal para mapear de un supervector GMM a un espacio expandido SVM. Con este proceso únicamente se tiene que hacer un producto interno entre el modelo deseado y el supervector GMM para obtener el resultado.

1.5 Década 2010's

En el año 2013 se realizó una solución de autenticación por medio de la voz para dispositivos android se utilizaron los coeficientes MFCC con venctores de 20 dimensiones [12], en la parte de entrenamiento se utilizó la cuantización vectorial una vez extraídas las características de la voz, se utilizó un lumbral para determinar si se escoge o se rechaza al hablante, basado en la distancia euclidiana. Para los experimentos se usaron diferentes cantidades de filtros y centroides para representar las señales, que provenían de los corpus Sphinx y GREYC, esto con la finalidad de disminuir la cantidad de datos que se estarían almacenando en el dispositivo.

En le extracción de los coeficientes MFCC se realizó una optimización en el número de filtros Mel y centroides a utilizar. Con ambas variaciones los errores variaron desde 4% hasta 6%. Dichos experimentos demuestran que no por tener más o menos parámetros se obtendrán mejores resultados y depende de más factores.

En 2015 se realizó una implementación de reconocimiento de voz y hablante para la automatización de un hogar [13], se colocaron audífonos en las diferentes habitaciones para capturar las señales de voz. En la parte de reconocimiento del hablante se utilizaron los coeficientes MFCC para la extracción de información de las grabaciones y el algoritmo GMM-UBM para la identificación, el cual es entrenada mediante Expectation-Maximization.

Para hacer más eficiente el sistema, a los datos ya entrenados en UBM se les realizó una adaptación usando el algoritmo Maximum A Posteriori , este es similar al Expectation-Maximization la diferencia radica en que utiliza los datos antiguos de UBM para combinarlos y crear nuevos parámetros. Por último, para identificación se utilizó un sistema basado en Semantic Classification Trees, donde el hablante que tuviera el valor más alto sería el seleccionado como correcto.

El corpus utilizado contó con 7803 grabaciones de 24 diferentes personas. Para las pruebas se utilizaron 11 hablantes, a los cuales se les pidió leer una serie de comandos y un artículo de periódico. Cada una de estas grabaciones variaba desde los 50 segundos, hasta los 300.

Una de las tecnologías más explotadas en los últimos años han sido las redes neuronales, en el año 2018 se presentó uno de los trabajos hechos en reconocimiento de hablantes haciendo uso de esta técnica [14]. En el primer paso para esta tarea, se crea el modelo universal basado en las características extraídas gracias a las redes neuronales profundas. Posteriormente las características particulares de cada hablante se utilizaron como entrada para el modelo generado previamente. Por último, para la evaluación se utiliza el algoritmo de similitud coseno, que se encarga de comparar le entrada contra el modelo del hablante seleccionado. VoxCeleb fue utilizado para los experimentos de este sistema, el cual consta con 140000 instancias de 1211 hablantes. Los audios se extrajeron de vídeos subidos a YouTube, cuentan con la misma cantidad de hombres y mujeres. Las características se extrajeron en tramos de 25 ms con traslape de 10 ms.

Se utilizaron diferentes técnicas para la creación de los modelos, una fue GMM-UBM, otra siendo i-vectores con y sin análisis discriminante lineal. Los experimentos que mejores resultados arrojaron fueron aquellos que utilizaron las redes neuronales para extraer las características no directamente de los archivos de audio, sino sobre las características extraídas por un método tradicional como podría ser MFCC.

1.6 Corpus Valquiria

El corpus utilizado para las pruebas del sistema de esta tesis, fue recolectado de llamadas telefónicas desde un teléfono público a celular, de público a teléfono fijo, celular a celular, celular a fijo y de fijo a fijo. Por lo tanto, no son grabaciones con un ambiente controlado, ya que contarán con todo el ruido ambiental generado durante cada una de las grabaciones.

Se grabaron personas en los siguientes rangos de edades: de 18 a 30, de 31 a 45, de 46 a 60 y de 60 en adelante. En cada uno de estos rangos se grabaron a 7 mujeres y 7 hombres diferentes, tomando en cuenta los 5 diferentes tipos de llamadas realizadas, dan un total de 280 grabaciones en el corpus, todas de diferentes personas.

A los participantes se les pidió leer un cuento llamado la cigarra y la hormiga y al finalizar se les realizaron algunas preguntas, con la finalidad de que hablaran de una forma más relajada. Como las personas leen diferente, la duración varia por cada uno, el promedio va de los 2 a los 3 minutos con las respuestas a las preguntas ya incluidas.

El "Corpus valquiria" fue diseñado por el "laboratorio de tecnologías del lenguaje" de la Facultad de Ingeniería y el grupo de ingeniería lingüística del Instituto de Ingeniería, ambos de la UNAM, durante los años 2015 y 2016. Este corpus aún no ha sido liberado al público, ya que se están realizando sus transcripciones fonéticas; las transcripciones antes referidas no son utilizadas en esta tesis.

Si se contara con un corpus grabado en mejores condiciones y en un ambiente controlado donde el ruido ambiental fuera menor, podría generar mejores resultados ya que se puede obtener mejores características de la voz de las distintas personas. En cambio, con un corpus de este estilo, se usan condiciones de uso más cotidiano y nos da una mejor idea de cómo se puede comportar el sistema. En la mayoría de los casos prácticos las grabaciones no van a ser obtenidas en un entorno controlado.

Los nombres de los archivos tienen un formato que permite identificar las características del hablante como lo son el género, rango de edad, el tipo de llamada y al final un índice que va del 0001 al 0007. A continuación, se listan las diferentes formas de identificar cada una de estas.

- Género: 01 para hombres, 02 para mujeres
- Rango de edad: "a" corresponde de 18 a 30 años, "b" de 31 a 45, "c" de 46 a 60 y "d" de 60 en adelante.
- Tipo de llamada: 01 calle a celular, 02 calle a fijo, 03 celular a celular, 04 celular a fijo y 06 fijo a fijo

Por ejemplo, si el nombre del archivo es el siguiente 02-a-04-0003svr.wav, significa que se trata de una mujer, de edad entre 18 y 30 años, la llamada se grabó de celular a fijo y es la tercera persona en esta categoría de las 7 presentes.

1.7 National Institute of Standards and Technology (NIST)

Un corpus diseñado para realizar pruebas en los sistemas de reconocimiento de hablantes independientes de texto es el realizado por National Institute of Standards and Technology (NIST) [15]. El cual fue recolectado por el Linguistic Data Consortium (LCD), está constituido por conversaciones telefónicas obtenidas a las afueras de américa del norte en idioma tagalo, cantonés, cebuano, mandarín, entre otros. A los sujetos que fueron parte de las grabaciones se les pidió que hablaran durante 10 minutos sobre el tema de su elección.

La evaluación del corpus consistió en utilizar una conversación como entrada y compararla contra los 12 idiomas que se tenían.

En cada prueba realizada el sistema entregaba dos resultados: el primero indicada si el idioma se encontraba entre los datos contra los que se comparaban: el segundo era un valor el cual indicaba que tan probable era que el idioma de entrada coincidiera con los idiomas contra los que se ejecutaba la prueba.

En el año 2003 se desarrolló un nuevo plan llamado Speaker Recognition Evaluation (SRE) [26], que consiste en 120 horas de conservaciones telefónicas en idioma inglés. La evaluación consistió en tres tareas:

- Detección de un hablante – datos limitados: grabaciones de 2 minutos fueron usadas para el entrenamiento, para las pruebas se usaron segmentos de una conversación de 1 minuto.
- Detección de dos hablantes – datos limitados: se usaron 3 conversaciones completas para cada hablante y conversaciones de 1 minuto para las pruebas.
- Detección de un hablante – datos extendidos: la diferencia entre esta tarea y la de datos limitados es que se utilizaron grabaciones de los hablantes de hasta 1 hora de duración.

1.8 Corpus AHUMADA

Es un corpus creado en España por la Universidad Politécnica de Madrid [16], cuenta con un total de 104 hombres grabados a lo largo de 6 sesiones. Cada una de dichas sesiones fueron realizadas con un intervalo de 11 días entre cada una.

Se tomaron algunas consideraciones para realizar las grabaciones:

- Lectura a diferentes velocidades de habla.
- Lectura contra habla espontánea.
- Diferentes teléfonos y micrófonos.
- Variaciones dialectales.

A los participantes del corpus se les pidió realizar una serie de enunciados:

- 2 repeticiones de los dígitos 0 al 9.
- 10 cadenas de dígitos con 10 dígitos cada una.
- 10 frases con 8 a 10 palabras.
- 1 texto con alrededor de 180 palabras.
- 2 repeticiones del mismo texto, una leyendo más rápido y otra más lento.
- 1 texto diferente cada sesión y diferente para cada hablante.
- Más de 1 minuto de una grabación libre, dónde se les pidió a los hablantes describieran algo familiar a ellos.

Los rangos de edad utilizados fueron tomados desde los 11, hasta más de 90. Se dividieron cada 10 años, así el primer intervalo va de 11 a 20 y así hasta llegar a 90 o más.

Capítulo 2. Procesamiento de la señal

2.1 Características de la voz

La voz es muy característica de cada persona y depende de diferentes órganos del cuerpo para su generación; a pesar de las diferencias en nuestras voces hay rasgos que se mantienen comunes cuando pronunciamos una misma palabra.

Por ejemplo, tomando el caso en que dos personas pronuncian la palabra “casa” y comparando sus espectrogramas podemos darnos cuenta de cómo la energía producida por cada individuo es diferente.

Los archivos utilizados en los siguientes ejemplos fueron grabados con un celular a una frecuencia de 8000 Hz y en formato WAV, para la gráfica, en el eje de las abscisas se encuentra el tiempo en milisegundos y en el eje de las ordenadas se encuentra la frecuencia.

En la figura 2.1 se puede observar claramente que se generó una mayor cantidad de energía al momento de pronunciar la palabra y en el caso de la figura 2.2 hay menos energía y se aprecia que la pronunciación de la última “a” fue bastante débil.

Incluso una persona puede generar espectrogramas diferentes dependiendo del estado de ánimo que tenga, ya que el tono de voz cambia con una emoción diferente. El ejemplo de la figura 2.3 muestra como se ve cuando la palabra se dice estando feliz y en la figura 2.4 estando enojado. Las diferencias en la cantidad de energía producida son muy notorias, ya que se puede apreciar claramente que cuando se pronunció enojado, las áreas correspondientes a la pronunciación de la “c” y la “a” están más obscuras que en comparación de la otra gráfica.

Aunque ambas grabaciones provienen de la misma persona, es claro que no siempre se va tener el mismo comportamiento en el tono de voz utilizado y para el reconocimiento de hablantes esto es de suma importancia, debido a que entre más grande sean los datos de entrenamiento, se puede abarcar una mayor cantidad de posibilidades.

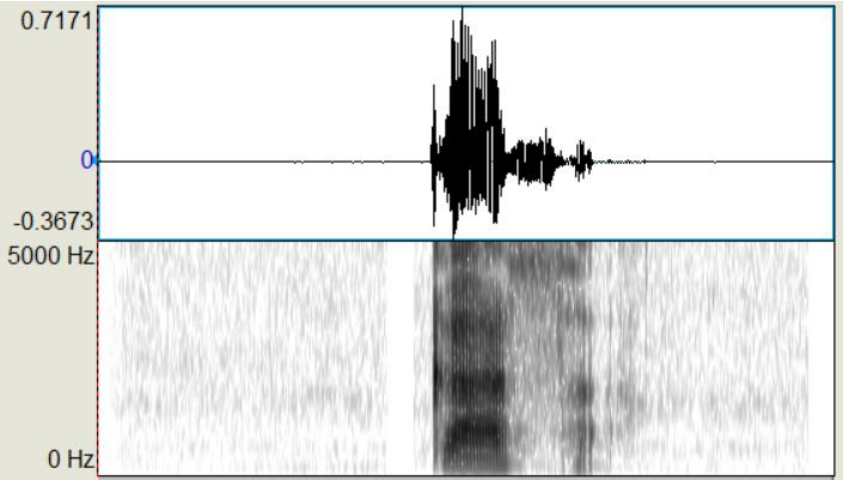

Figura 2.1. Persona 1 diciendo casa.

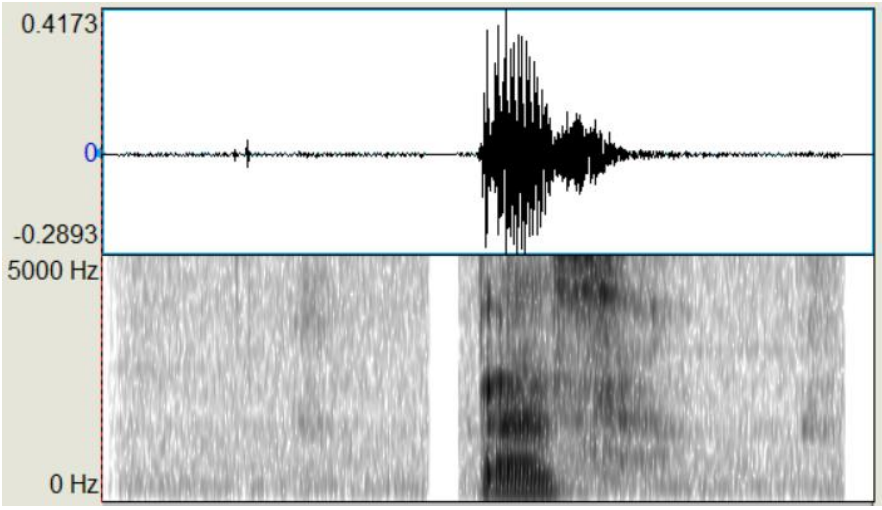

Figura 2.2. Persona 2 diciendo casa.

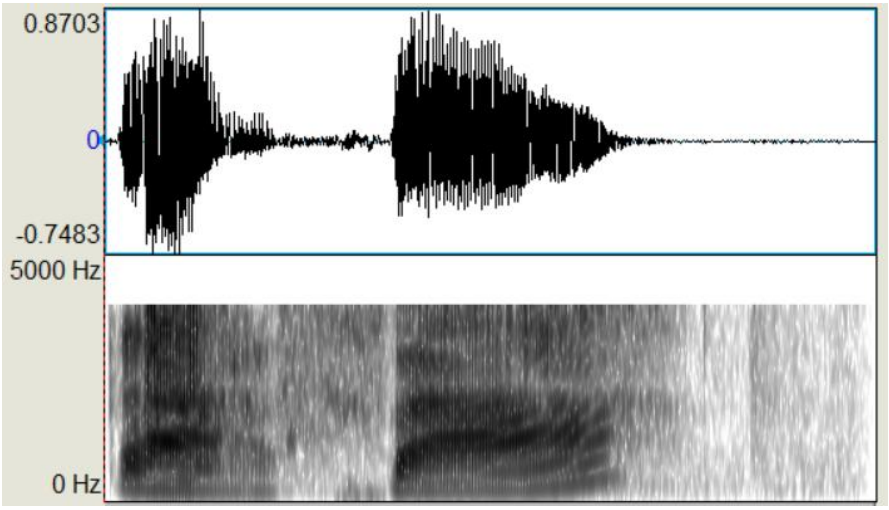

Figura 2.3. Palabra casa en estado feliz.

Incluso se pueden apreciar las diferencias cuando se produce un sonido sordo con una intensidad diferente; en la figura 2.3 al producirse el sonido de la “s” se nota en el espectrograma que tiene una cantidad muy baja de energía y en la gráfica de la señal en tiempo tiene muy poca variación, en cambio, en la figura 2.4 hay una mayor generación de energía y en consecuencia, se ven más partes obscuras en el espectrograma; por otro lado, en la otra gráfica 2.3 se ve que hubo mayor variación en la señal al pronunciar la “s”.

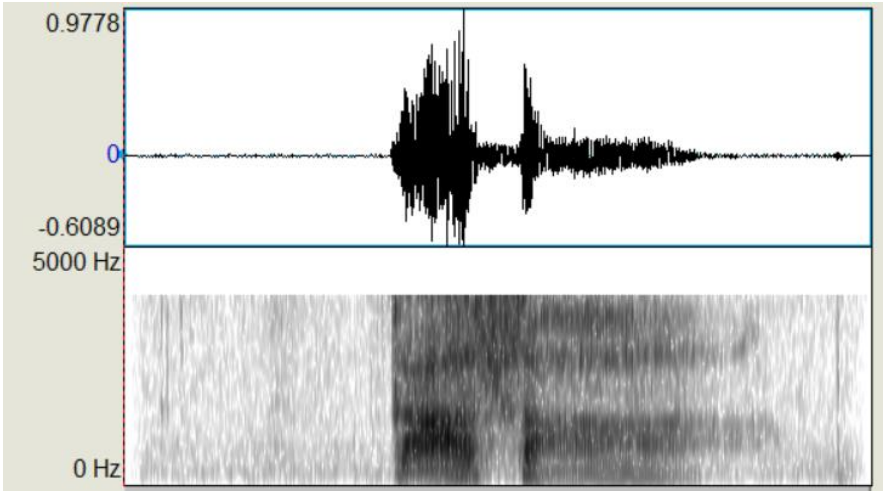

Figura 2.4. Palabra casa de manera enojada.

Debido a que en el reconocimiento de hablantes las emociones son un factor importante que siempre se debe de tener en cuenta, vamos a analizar el comportamiento de la voz de una persona diciendo la palabra “manzana” estando normal, feliz, enojado y triste.

En la figura 2.5 se muestra la señal en tiempo y el espectrograma de la persona con un estado de ánimo cotidiano, que servirá de referencia para ver los cambios producidos al decirse lo mismo con diferentes emociones.

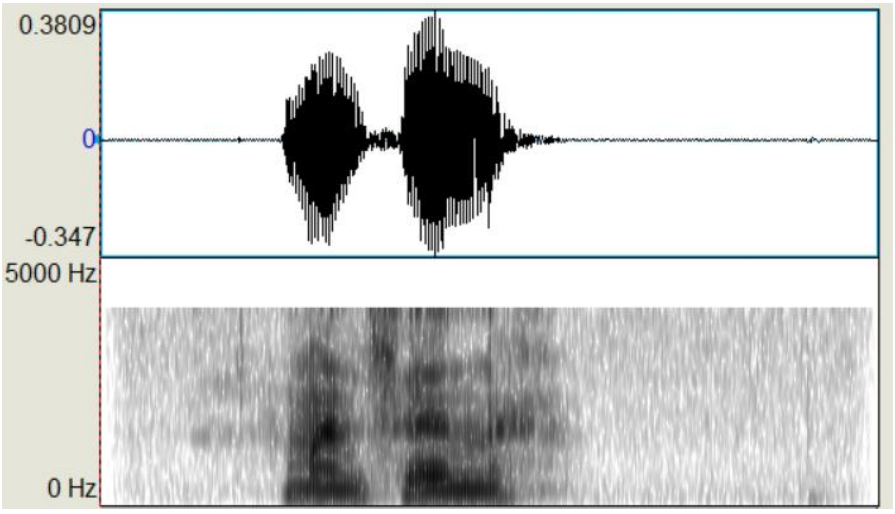

Figura 2.5. Palabra manzana en estado cotidiano.

Estando feliz la gráfica resultante se muestra en la figura 2.6.

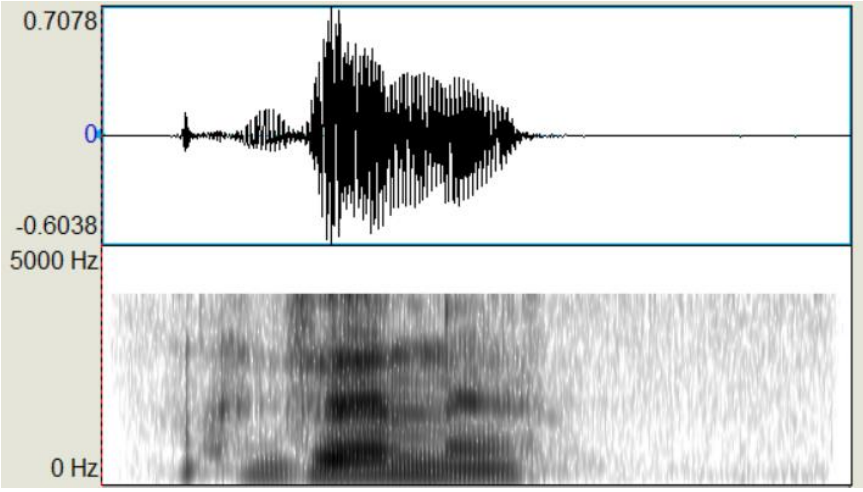

Figura 2.6. Palabra manzana en estado feliz.

Al comparar ambas señales, se aprecia fácilmente que las diferencias son muy obvias. Estas se pueden ver cuando se pronuncia la silaba “za”, ya que la generación de energía fue mayor cuando se dijo estando feliz y se muestra en el espectrograma porque tiene un sombreado más obscuro.

Ya que los sonidos sordos generan una menor energía, al decirlo feliz nuestra voz se suele escuchar más fuerte y por ende se genera más energía en ese tipo de ruidos.

En el caso de una persona hablando enojada y diciendo la palabra “manzana” se obtienen los resultados mostrados en la figura 2.7.

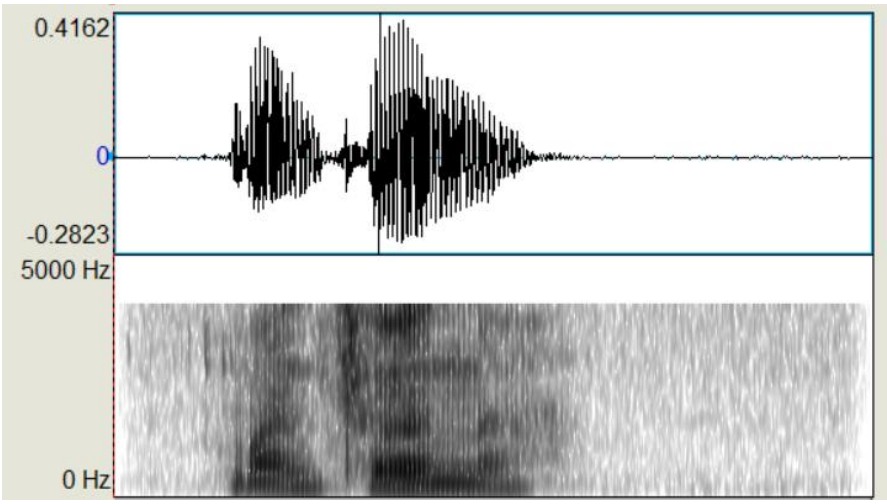

Figura 2.7. Palabra manzana en estado enojado.

Una vez más se tienen resultados que difieren de los anteriores. Estando enojado se observa claramente que la mayor concentración de energía es en las partes sonoras y la no sonora no tuvo el mismo nivel que se obtuvo cuando se dijo la palabra estando feliz.

Por último, cuando se dice la palabra con un estado de tristeza se genera la gráfica en la figura 2.8.

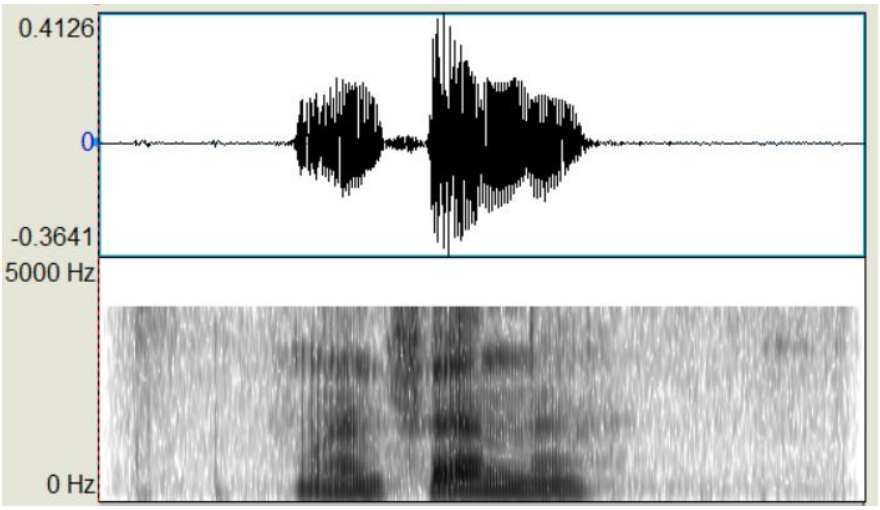

Figura 2.8. Palabra manzana en estado triste.

En este último caso se puede ver que la energía durante toda la palabra fue muy baja, en comparación con el resto de las señales anteriores. Esto se puede apreciar igualmente cuando se escucha el audio utilizado, la voz se escucha menos fuerte que las anteriores.

Con estas 4 variaciones en las emociones al pronunciar una palabra, es evidente que cada una es diferente, aún siendo la misma palabra y persona, los espectrogramas generados muestran niveles de energía diferentes según la forma en que se pronuncia.

Existen personas que tienen un tono de voz similar cuando las escuchamos hablar, pero ¿Qué pasa cuando se muestra la señal y el espectrograma? Para esta comparación se utilizan dos grabaciones de dos mujeres diciendo la palabra “escalera”.

En la figura 2.9 y 2.10 se pueden apreciar las gráficas generadas a partir de los dos archivos de prueba. Como se puede observar, en el primer espectrograma la cantidad de energía generada por la persona fue mayor que la otra. Gracias a esto podemos darnos cuenta de que los sistemas computacionales pueden obtener información que permite ver que las diferencias son más de las que creíamos en las características de la voz, a diferencia de nuestro oído, para el cual puede sonar muy similar una voz con otra.

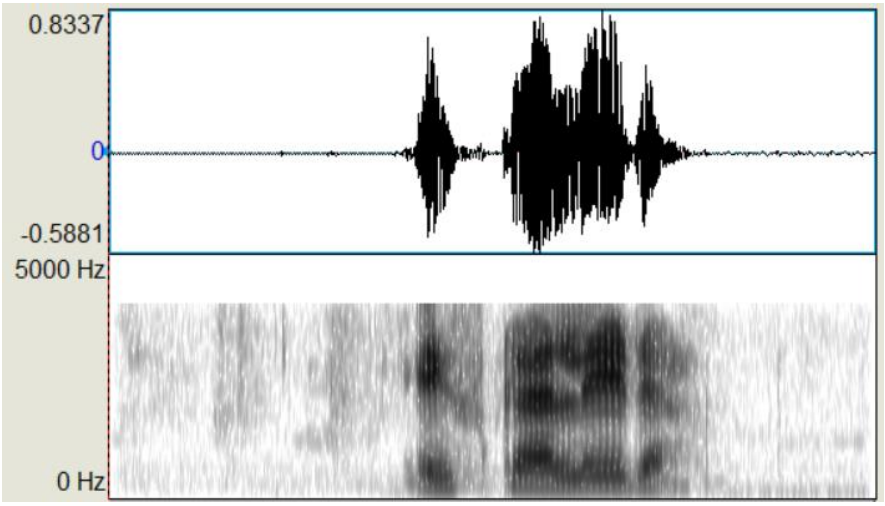

Figura 2.9. Palabra escalera por mujer 1.

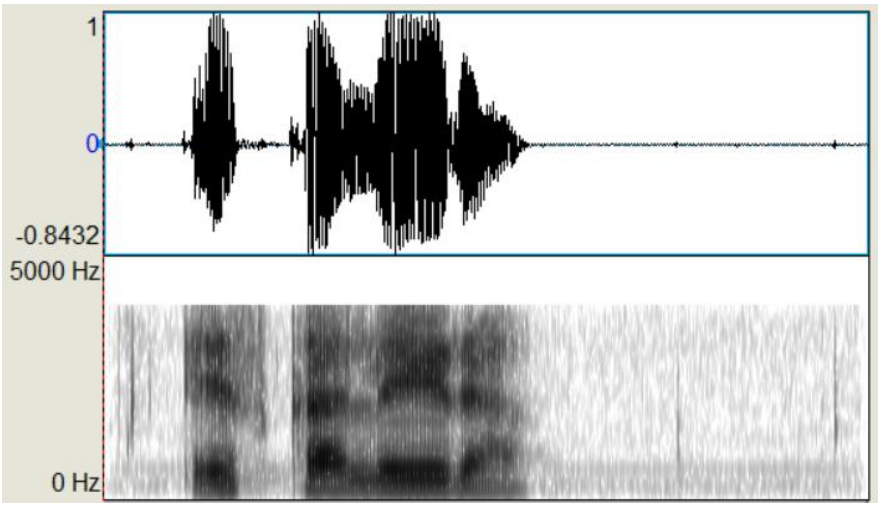

Figura 2.10. Palabra escalera por mujer 2.

2.2 Análisis de una señal de voz

Para realizar cualquier aplicación de reconocimiento de voz primero hay que hacer una serie de pasos para poder trabajar con las señales de voz, dichas que son capturadas por el usuario a través de un micrófono. La figura 2.11 muestra un diagrama de bloques de los primeros pasos del proceso a seguir en el análisis de la señal de voz.

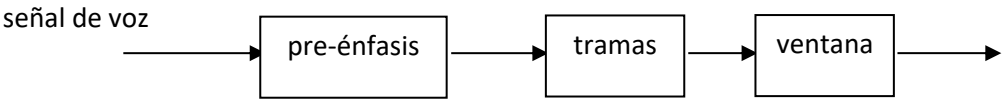

Figura 2.11.

El primer paso en el análisis de la señal es pasarla por un filtro pre-énfasis. Con esto se busca aumentar las frecuencias altas para tratar de compensar las partes que fueron suprimidas durante la generación del sonido. El cálculo de dicho filtro se realiza mediante la ecuación:

$$y(n) = x(n) - ax(n-1)$$

Donde *a* es una constante entre los valores 0.9 y 1.0.

Una vez se tiene la señal con el filtro aplicado, se dividen en tramas de *N* muestras toda la señal, de esta forma el análisis podrá ser más preciso al trabajar individualmente con cada una de estas tramas. Al realizar esta segmentación de la señal, se pueden presentar discontinuidades entre los inicios y finales de las tramas de la señal original al procesarlas, una forma de solucionar este problema es hacer uso del traslape.

El traslape quiere decir que cada trama contendrá un segmento de la trama anterior, típicamente se utiliza 25% o 50%. Lo que se logra implementando esta técnica es evitar esas pérdidas en las tramas, ya que al haber segmentos con partes de la anterior se puede mantener la señal original sumando los traslapes entre las divisiones de las tramas.

El último paso a realizar antes de empezar con la extracción de las características es aplicar una ventana a las tramas resultantes con el fin de evitar más discontinuidades en las transiciones de las tramas. La ventana utilizada comúnmente es la de Hamming, definida como:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \quad 0 \leq n \leq N-1$$

Después de haber realizado todos los pasos anteriores se puede proceder a la extracción de características de la señal de voz, la cual es fundamental en el desarrollo ya que, utilizando una técnica adecuada, puede generar mejores resultados. Dichas características pueden obtenerse tanto en el domino del tiempo o en un dominio diferente, según sea el análisis que se vaya a aplicar.

Para esto utilizaremos los coeficientes MFCC (Mel-Frecuency Cepstral Coefficients)[17] [18] [19] los cuales son una representación definida como el cepstral real de una señal en tiempo corto derivada de la transformada de Fourier de la señal [1]. Se conoce como un proceso homomorfico ya que convierte la convolución definida en la ecuación 2.1 en la suma dada por ecuación 2.2.

$$x[n] = e[n] * h[n]$$
Ecuación 2.1

$$\hat{x}[n] = \hat{e}[n] + \hat{h}[n]$$
Ecuación 2.2

El modelo de la producción humana de la voz adoptado es el modelo fuente-filtro y gracias a esta deconvolución nos permite separar la fuente del filtro, en otras palabras, el pulso de la glotis y el impulso del tracto vocal. La fuente está relacionada con el aire expulsado por los pulmones, al momento de generar la voz si el sonido emitido es sordo, como es el caso de la "s" y la "f", la glotis se abre y las cuerdas vocales se relajan, si el sonido es sonoro las cuerdas vocales vibran. Con esta idea podemos separar y quedarnos únicamente con el filtro, de esta forma podremos crear un modelo matemático que describa el comportamiento del tracto vocal.

Se busca computar un análisis de frecuencia basado en un banco de filtros, en la escala mel se cuenta con un espaciado lineal para frecuencia menores a 1000 Hz y un espaciado logarítmico para frecuencias superiores.

El primer paso es obtener la transformada rápida de Fourier (FFT) de cada una de las tramas obtenidas previamente y así poder convertirlas del dominio del tiempo al dominio de la frecuencia. Posteriormente, nos quedamos únicamente con el algoritmo de la amplitud del espectro y se descarta la información de la fase, debido a que se ha demostrado en estudios que la amplitud es de mayor importancia.

El rango de frecuencias en el espectro FFT es muy amplio y las señales de voz no siguen una escala lineal. En la figura 2.12 se muestran los filtros triangulares que se usan para computar la suma de las componentes espectrales del filtro. Cada filtro tiene una respuesta paso banda, el espaciado como el ancho de banda están determinados por un intervalo constante.

Este banco de filtros se ha diseñado para emular el filtrado paso banda que se cree que ocurre en el sistema auditivo.

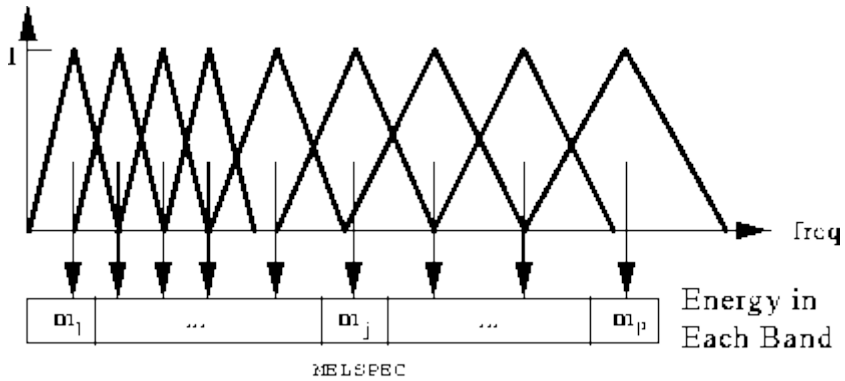

Figura 2.12. Banco de filtros en escala Mel

La repuesta de cada filtro tiene forma triangular e igual a la unidad en la frecuencia central y decrece linealmente a la frecuencia central de los dos filtros adyacentes. Así la salida de cada filtro es la suma de sus componentes espectrales filtradas. Posteriormente, se utiliza la siguiente ecuación para computar la escala mel para la frecuencia f dada en Hz.

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

Como último paso se busca regresar de la escala mel al dominio del tiempo, dado que los coeficientes en la escala mel y su logaritmo, son números reales podemos transformarlos al dominio del tiempo utilizando la transformada discreta de coseno (DCT). Al resultado de esta última transformación se le conoce como los coeficientes MFCC.

En la figura 2.13 se observa el diagrama de bloques de los últimos pasos para obtener los coeficientes.

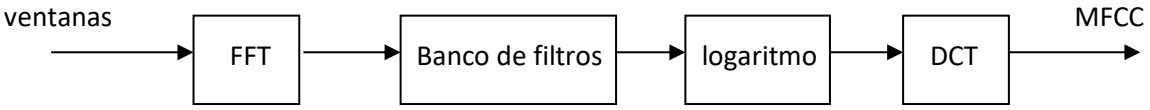

Figura 2.13. Diagrama de bloques para extracción de coeficientes MFCC.

Capítulo 3. Gaussian Mixture Model (GMM) y Maximum Likelihood Estimation (MLE)

3.1 Gaussian Mixture Model (GMM)

GMM [6][10][23] es un modelo que utiliza la función densidad de probabilidad representada como una suma con pesos de componentes de densidad gaussiana. Es ampliamente utilizado para modelar la probabilidad de distribución de mediciones continuas, o como en este caso, características biométricas. Cuando se habla de reconocimiento de hablantes se utilizan los espectros, debido a que refleja la estructura del tracto vocal de cada persona, el cuál es el factor principal que sirve para poder diferencial la voz de una persona y otra.

La suma de las componentes para GMM está dada por la ecuación

$$p(\vec{x}|\lambda) = \sum_{i}^{M} w_i b_i(\vec{x}) \ldots\ldots (1)$$

Donde $\vec{x}$ es un vector de entrada de dimensión D con los valores correspondientes a la señal de voz a analizar, $b_i(\vec{x})$ son las componentes de densidad del modelo, w_i son los pesos de cada componente, los cuales satisfacen la siguiente restricción $\sum_i^M w_i = 1$. Las componentes de densidad se obtienen mediante la ecuación 2 que es una función gaussiana.

$$b_i(\vec{x}) = \frac{1}{2\pi^{1/2}|\Sigma_i|^{1/2}} exp\left\{-\frac{1}{2}(\vec{x} - \vec{\mu}_i)' \sum_{i}^{-1}(\vec{x} - \vec{\mu}_i)\right\} \ldots\ldots (2)$$

$\vec{\mu}_i$ es el vector de medias y Σ_i la matriz de covarianza.

Los parámetros utilizados por el modelo GMM se representan de la forma

$$\lambda = \{w_i, \vec{\mu}_i, \Sigma_i\} \quad i = 1, \dots, M$$

Para realizar la estimación de los parámetros que nos darán el modelo con el que se representarán los hablantes, se utiliza el algoritmo Expectation-Maximization (EM). Consiste en un proceso aleatorio que parte de un modelo aleatorio λ inicial para obtener uno nuevo $\bar{\lambda}$, de manera que se cumpla $p(X|\bar{\lambda}) > p(X|\lambda)$, posteriormente el nuevo modelo se vuelve el inicial para la siguiente iteración y el algoritmo se detendrá hasta que haya una convergencia en las probabilidades o se alcance el límite en el número de iteraciones especificadas. En cada una de las iteraciones del algoritmo se llevan a cabo dos pasos, el primero es el E y después se realiza el M.

El paso E consiste en estimar los pesos que tiene cada uno de los puntos en $\vec{x}$ con todas las componentes de densidad λ_k, con el fin de saber la probabilidad de que un punto haya sido generado por una componente u otra, para realizar el cálculo se hace con la siguiente ecuación

$$w_{ik} = \frac{w_i b_i(\vec{x}_t)}{\sum_{k=1}^{M} w_k b_k(\vec{x}_t)} \dots\dots (3)$$

En el paso M se hace uso de los pesos obtenidos en el paso E para obtener los nuevos valores de los parámetros necesarios para el modelo, las ecuaciones de actualización son las siguientes:

Para los pesos de las componentes

$$\bar{w}_k = \frac{1}{T}\sum_{i=1}^{T} w_{ik} \dots\dots (4)$$

Para las medias

$$\vec{\bar{\mu}}_k = \frac{\sum_{i=1}^{T} w_{ik}\vec{x}_i}{\sum_{i=1}^{T} w_{ik}} \dots\dots (5)$$

Las matrices de covarianza

$$\bar{\Sigma}_{\mathrm{k}} = \frac{\sum_{i=1}^{T} w_{ik}(\vec{x}_i - \bar{\mu}_k)(\vec{x}_i - \bar{\mu}_k)'}{\sum_{i=1}^{T} w_{ik}} \dots\dots (6)$$

Para calcular el log likelihood se hace uso de la ecuación

$$l(\lambda) = \sum_{k=1}^{K} \log\left(\sum_{i=1}^{T} w_{ik} p(\vec{x}_i|\lambda_k)\right) \dots\dots (7)$$

Cuando se haya cumplido el criterio de convergencia los valores finales serán los utilizados para el modelo del hablante en cuestión.

3.2 Maximum Likelihood Estimation (MLE) y Curvas ROC

El concepto básico sobre este algoritmo es tomar la decisión sobre cuál es la fuente más parecida al fenómeno analizado, en términos de reconocimiento del hablante, seleccionar al hablante que sea más parecido al que se está comprobando en ese instante [20][21][22]. Requiere únicamente saber la función densidad de probabilidad condicional de cada observación dada y los posibles candidatos, esto es, $p(X|\lambda_1)$ hasta $p(X|\lambda_k)$, donde X es el hablante a identificar y λ_k corresponde a los modelos de hablantes almacenados.

Para realizar la identificación dentro de un grupo de modelos S representados por los GMM's $\lambda_1, \dots, \lambda_S$, lo único que se debe de hacer es encontrar el modelo que maximice la probabilidad dada por la ecuación

$$\hat{S} = arg \max_{1 \le k \le S} p(X|\lambda_k) \dots . . (8)$$

Utilizando valores logarítmicos se puede reescribir como

$$\hat{S} = arg \max_{1 \le k \le S} \sum_{t=1}^{T} \log p(\vec{x}_t|\lambda_k) \dots . . (9)$$

Donde $p(\vec{x}_t|\lambda_k)$ está dada por la ecuación (1).

Consideremos un problema de predicción de clases binario, en la que los resultados se etiquetan positivos (p) o negativos (n). Hay cuatro posibles resultados a partir de un clasificador binario como el propuesto. Si el resultado de una exploración es p y el valor dado es también p, entonces se conoce como un Verdadero Positivo (VP); sin embargo, si el valor real es n entonces se conoce como un Falso Positivo (FP). De igual modo, tenemos un Verdadero Negativo (VN) cuando tanto la exploración como el valor dado son n, y un Falso Negativo (FN) cuando el resultado de la predicción es n pero el valor real es p.

Una curva ROC (Receiver Operating Characteristic) es una representación gráfica de la sensibilidad frente a la especificidad para un sistema clasificador binario según se varía el umbral de discriminación. Otra interpretación de este gráfico es la representación de la razón o ratio de verdaderos positivos frente a la razón o ratio de falsos positivos también según se varía el umbral de discriminación (valor a partir del cual decidimos que un caso es un positivo).

La tabla de contingencia puede proporcionar varias medidas de evaluación. Para dibujar una curva ROC sólo son necesarias las razones de verdaderos positivos y de falsos positivos. Los verdaderos positivos miden hasta qué punto un clasificador o prueba diagnóstica es capaz de detectar o clasificar los casos positivos correctamente, de entre todos los casos positivos disponibles durante la prueba. Los falsos positivos definen cuántos resultados positivos son incorrectos de entre todos los casos negativos disponibles durante la prueba.

La representación obtenida por este método tiene forma aproximadamente en escalera. En efecto, para cada variación mínima del valor de corte que produzca cambios en sensibilidad o especificidad, al menos un caso pasa a ser considerado bien como verdadero positivo, lo que corresponde con un trazo vertical o bien como falso positivo, lo que da lugar a un trazo horizontal.

En la figura 3.1, se muestran 3 ejemplos de curvas que se pueden obtener. Si la curva se va aproximando cada vez más a una diagonal, esto significará que los resultados obtenidos en los experimentos no fueron buenos y hubo muchos errores. En caso contrario, la curva va siendo más prominente, será indicativo de que los resultados son mejores.

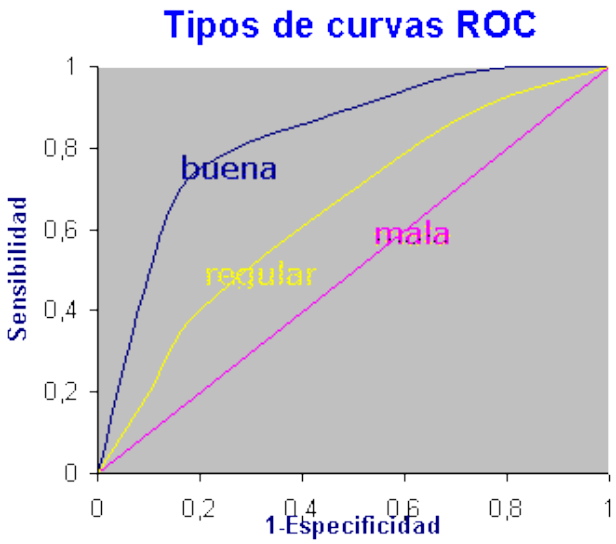

Figura 3.1. Indicadores de resultados en curva ROC

3.3 Ejemplo

Tomando un vector de dimensión 1110 con números aleatoriamente generados, podemos probar el algoritmo y de esta forma probar que no solo obtiene buenos resultados, aunque no se trate de una señal de voz. En la figura 3.2 se muestra la gráfica de los valores a los que se les aplicará GMM.

Al aplicar el algoritmo con 10 componentes gaussianas y un límite de 10 iteraciones, nos encontramos con que el algoritmo converge en la séptima iteración. En la figura 3.3 se puede observar el histograma de los datos en color gris y las componentes con diferentes colores, la suma de estas, que es el modelo final, se observa en la línea roja más gruesa.

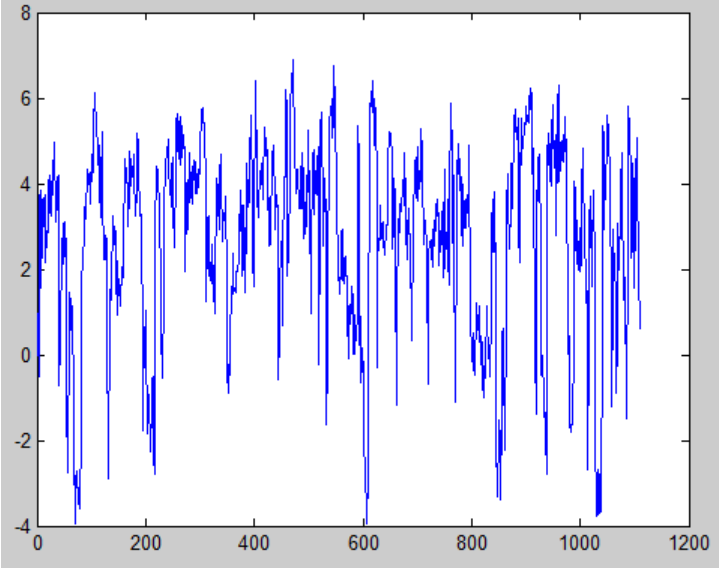

Figura 3.2.

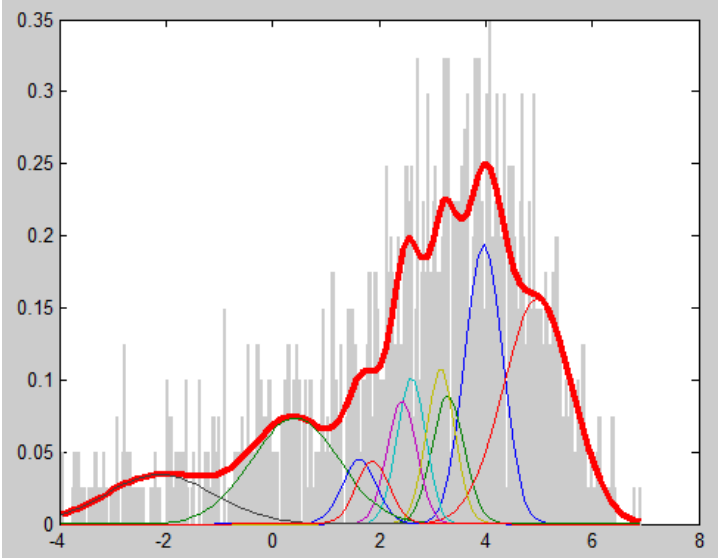

Figura 3.3.

Tomando en cuenta un archivo de voz de una palabra en formato WAV, se dividirá en tramas de 30 ms, cada una estará representada por 5 coeficientes.

El primer paso que se debe hacer es leerlo y obtener los valores en el dominio del tiempo que representan dicha señal. La señal de voz se almacena en un vector, como el que se muestra a continuación, que más adelante se utilizará para el cálculo de los demás valores.

0.0021	0.0012	0.0002	-0.0002	-0.0007	…	-0.0036	-0.0026	-0.0021	-0.001

Posteriormente, se debe calcular los coeficientes MFCC que servirán para representar la señal de voz y realizar el algoritmo GMM, para obtenerlos hay que mandar el vector con los valores de la señal y la función *melcepst* regresará una matriz con la cantidad de coeficientes especificados. Teniendo en cuenta que la señal de voz se dividirá en tramas, cada una de estas estará representada por la misma cantidad de coeficientes, lo cual dará como resultado una matriz de M tramas por N coeficientes. La señal anterior arrojará los siguientes coeficientes.

Tramas	Coeficiente 1	Coeficiente 2	Coeficiente 3	Coeficiente 4	Coeficiente 5
Trama 1	1.0893	0.2954	0.7190	-0.3447	-0.6857
Trama 2	1.3151	0.2524	0.5422	-0.6759	-0.5343
Trama 3	1.4705	0.4910	-0.1482	-1.4290	-0.8262
Trama 4	0.3129	0.2535	0.1315	-1.1136	-1.5100
Trama 5	-0.0418	-0.0708	-0.1311	-0.1387	-0.3389
Trama 6	1.1786	0.3087	-1.2927	-1.3329	-1.0388
Trama 7	1.0928	0.0620	-0.0343	-0.4339	-0.5137
⋮	⋮	⋮	⋮	⋮	⋮
Trama 21864	0.3966	0.7068	0.3075	-0.4973	-1.3689

Tabla 3.1 Coeficientes MFCC.

Con la ayuda de estos coeficientes, ahora es posible empezar con el algoritmo GMM para obtener los parámetros que formarán parte del modelo matemático del hablante en cuestión. Primero se debe definir la cantidad de componentes gaussianas que se utilizarán en los modelos y el número de iteraciones que se van a realizar en caso de que el algoritmo no llegue a un estado de convergencia antes. En este ejemplo se utilizarán 4 componentes gaussianas y 10 iteraciones.

Como resultado obtendremos 3 vectores con la información generada por GMM, estos vectores contienen las medias, varianzas y los pesos, que son los parámetros utilizados en cada una de las componentes gaussianas que se sumarán para obtener el modelo final del hablante.

Para obtener los valores de Mu se hace uso de la ecuación número 5, sigma se obtiene con la ecuación 6 y los pesos de cada componente se usa la ecuación número 4. Los valores arrojados son los siguientes mostrados en las tablas, donde en cada columna se muestra el valor obtenido para

cada una de las componentes en el caso de mu y sigma, para los pesos los valores de cada componente se encuentran en las filas.

Mu

Gaussiana 1	Gaussiana 2	Gaussiana 3	Gaussiana 4
0.7541	4.0326	3.6634	0.8863
-0.0233	-0.7124	1.3232	-1.1682
0.2456	-0.9001	0.4567	-0.4144
-0.4834	-1.5370	-2.4563	-0.9959
-0.6211	-0.8291	-1.0846	-0.4512

Sigma

Gaussiana 1	Gaussiana 2	Gaussiana 3	Gaussiana 4
1.1311	2.8304	0.8245	5.9404
0.8310	1.7680	1.2601	1.9136
0.3437	1.2712	0.7967	0.7690
0.3352	0.6356	1.0147	0.6707
0.2847	0.5620	0.5684	0.4274

Pesos

Gaussiana 1	0.2909
Gaussiana 2	0.2172
Gaussiana 3	0.1831
Gaussiana 4	0.3086

Cuando se grafican estos parámetros se puede observar las siguientes gráficas, figura 3.4.

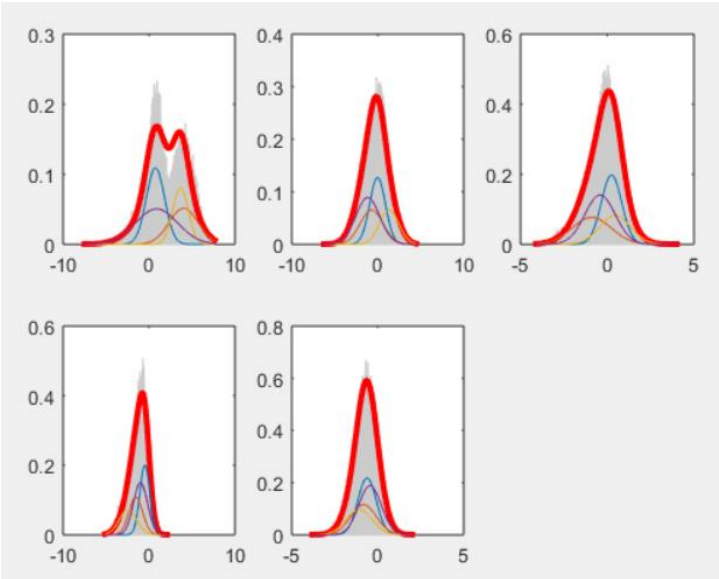

Figura 3.4.

Cada gráfica individual representa los 5 coeficientes de las tramas, en la parte gris se encuentra el histograma de los valores de densidad obtenidos a partir de la ecuación 2, las gaussianas chicas en colores son las que se obtuvieron con el algoritmo GMM y la más grande, la roja, es la suma de estas pequeñas gaussianas y representa el modelo final del hablante.

Capítulo 4. Programa principal

El programa con la implementación de la extracción de los coeficientes Mel-Frecuency Cepstrum Coefficient (MFCC) y del algoritmo Gaussian Mixture Model (GMM) fue desarrollado utilizando la herramienta Matlab y el toolkit voicebox, en la figura 4.1 se muestra la estructura de las funciones empleadas en el código.

Se recibe una señal de voz con la cual se va a realizar la prueba y otras señales con las que se va a comparar para determinar a cuál de ellas corresponde. El proceso empieza por calcular los coeficientes MFCC con la ayuda de la función *melcepst*, posteriormente se realiza el algoritmo GMM para obtener los modelos matemáticos correspondientes a cada hablante, finalmente se realiza la comparación de los coeficientes de la señal de prueba contra los modelos, finalmente para determinar al que corresponde se calcula el log-likelihood. Una vez obtenido este valor se muestra la matriz de confusión donde se observará que el valor más alto será del hablante reconocido.

El programa principal comienza definiendo variables que se van a utilizar al llamar algunas funciones, una es el número de gaussianas que se van a utilizar para realizar el algoritmo GMM, en este caso serán 4 y la otra es la frecuencia de muestreo de los archivos de audio a utilizar durante la prueba del experimento, tal como se muestra en la figura 4.2.

Se procede a leer los archivos para llevar a cabo la etapa de entrenamiento. Con la función *dir* de matlab se listan los archivos contenidos en la carpeta correspondiente. Creamos una variable para almacenar la ruta de la carpeta contenedora de las grabaciones de prueba. Por último, se obtiene la cantidad de archivos existentes. Figura 4.3.

Para empezar el procesamiento de los datos y obtener los modelos de los hablantes, se inicializa un ciclo for para ir recorriendo cada uno de los archivos. Con la función *audioread* se leen los archivos y se almacenan en una variable temporal I.

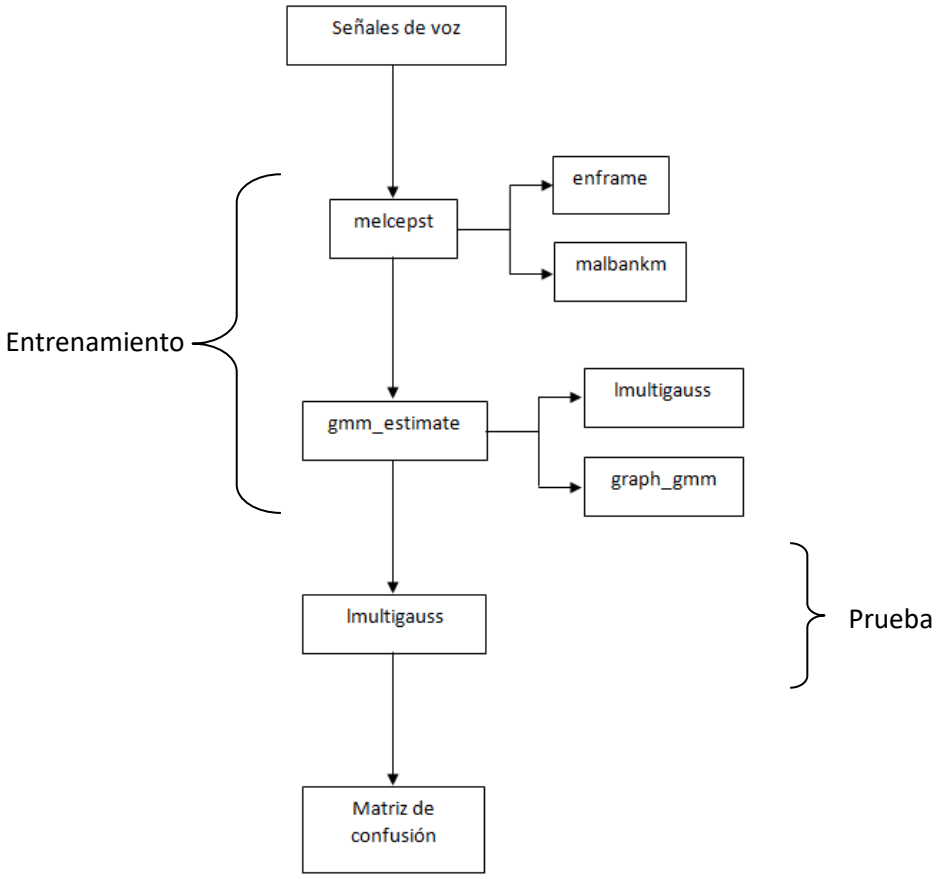

Figura 4.1. Diagrama de funciones utilizadas.

```
No_of_Gaussians=4;
Fs=8000;
```

Figura 4.2. Definición de variables.

```
read_train=dir('C:\Documentos\MATLAB\GMM\train\*.wav');
name_train='C:\Documentos\MATLAB\GMM\train\';
train=length(read_train);
```

Figura 4.3. Ubicación de los archivos para el entrenamiento.

El siguiente paso es la obtención de los coeficientes MFCC. La función *melcepst* recibirá 4 parámetros, que son la señal de voz almacenada en I, la frecuencia de muestreo Fs, el tipo de ventana que se le aplicará y por último la cantidad de coeficientes que se van a extraer. El resultado que arrojará será una matriz conteniendo los coeficientes MFCC por cada una de las ventanas resultantes de las señales de voz. Dichas matrices estarán almacenadas en una matriz celular para poder ser utilizados en el siguiente paso, que es la obtención de los modelos correspondientes.

El último paso se realiza con la función *gmm_estimate*, ésta recibirá 3 parámetros: los coeficientes MFCC, el número de gaussianas que se van a utilizar y el número de iteraciones que va a ejecutar el algoritmo. La salida del programa serán los parámetros con los que se va a modelar la voz de cada uno de los hablantes, se almacenarán en tres matrices celulares.

```
for k = 1:train
archivo = read_train(k).name;
I = audioread(strcat(name_train,archivo));
training_features{k}=melcepst(I,Fs,1,5);
[mu_train{k},sigma_train{k},c_train{k}]=gmm_estimate(training_featu
res{k}',No_of_Gaussians,3);
end
```

Figura 4.4. Lectura de archivos, obtención de coeficientes y modelo final.

La etapa de evaluación se realiza de forma similar a la de entrenamiento, se lee la carpeta contenedora de los archivos de voz. Posteriormente en un ciclo for se leen los archivos y después se realiza la obtención de los coeficientes mfcc, figura 4.5.

```
for k = 1:test
archivo = read_test(k).name; I =
audioread(strcat(name_test,archivo));
testing_features{k}=melcepst(I,Fs,1,5);
end
```

Figura 4.5. Archivos de prueba.

Ya con los coeficientes se procede a realizar la comparación contra los modelos obtenidos previamente. En orden de realizar la evaluación se hace un ciclo for anidado donde el primer for se encargará de recorrer los datos de evaluación y el segundo se encargará se recorrer cada uno de los modelos. Con la función *lmultigauss* se calcula el log-likelihood este valor determinará a que hablante corresponde la voz de entrada.

La forma de almacenar los datos es dentro de una matriz, donde las filas corresponden a las grabaciones de prueba y las columnas las grabaciones de entrenamiento. El programa determinará el log-likelihood de los datos de prueba contra todos los modelos y el valor máximo será el correspondiente a la coincidencia de la señal de prueba con el modelo más parecido.

```
for i=1:test
    for j=1:train
        [lYM,lY]=lmultigauss(testing_features{i}',
mu_train{j},sigma_train{j},c_train{j});
        B(i,j)=mean(lY);
    end
    [M,P] = max(B(i,:));
    x = sprintf('Voice %d corresponds to model %d',i, P);
    disp(x);
end
```

Figura 4.6. Etapa de evaluación.

4.1 Función melcepst

Para hacer uso de la función *melcepst*, hay que llamarla con los argumentos correctos que son el arreglo con los valores correspondientes a la señal de voz, el tipo de ventana a implementar y el número de coeficientes deseados, figura 4.7.

```
training_features1=melcepst(training_data1,Fs,1,3);
testing_features1=melcepst(testing_data1,Fs,1,3);
```

Figura 4.7. Llamada de la función melcepst.

Para el cálculo de los coeficientes primero hay que hacer un preprocesado en los archivos, el cual consiste en dividir en ventanas las señales de voz, con la función *enframe* se realizan esta tarea y nos regresará una matriz donde las columnas corresponderá a la cantidad de elementos que tenga la ventana y las filas será el número de ventanas que se obtienen de las señales. Se tiene que llamar la función con los 3 parámetros que recibe, la señal que se va a particionar (x), el tipo de ventana que se va a aplicar, que puede ser hamming, hanning o una rectangular en el dominio del tiempo (win) y el último corresponde a la separación que habrá entre el inicio de cada una de las ventanas (inc).

En la figura 4.8 está el código utilizado para llamar la funcion y que realice el cálculo de las ventanas que se estarán utilizando durante los siguientes pasos del algoritmo.

```
if length(w)==0
   w='M';
end
if any(w=='R')
   z=enframe(s,n,inc);
elseif any (w=='N')
   z=enframe(s,hanning(n),inc);
else
   z=enframe(s,hamming(n),inc);
end
```

Figura 4.8. Obtención de ventanas.

En la figura 4.9 se observan los comandos utilizados para poder obtener las ventanas acorde a los parámetros recibidos, los vectores generados por la función *enframe* serán guardados en una matriz, donde cada fila corresponde a los valores de una ventana y la cantidad renglones es igual al número de ventanas obtenidas. Lo primero que realizar el algoritmo es calcular el número de ventanas que va a tener y en caso de que al final los datos restantes no sean suficientes para llenar una ventana, se descartarán dichos valores.

Ya con la señal dividida en ventanas se termina el pre procesado y se empiezan los pasos para obtener los coeficientes.

El primero que se debe realizar es sacar la transformada discreta de Fourier (DFT) de cada una de las ventanas con la función *rfft*. El siguiente paso en la obtención de los coeficientes, es calcular un banco de filtros, que es un filtro paso-bajas, el cual ayuda a que solo manejemos la información correspondiente al tracto vocal. Para realizar este paso se utiliza la función *melbankm*, figura 4.10.

```
function f=enframe(x,win,inc)
nx=length(x);
nwin=length(win);
if (nwin == 1)
   len = win;
else
   len = nwin;
end
if (nargin < 3)
   inc = len;
end
nf = fix((nx-len+inc)/inc);
f=zeros(nf,len);
indf= inc*(0:(nf-1)).';
inds = (1:len);
f(:) = x(indf(:,ones(1,len))+inds(ones(nf,1),:));
if (nwin > 1)
    w = win(:)';
    f = f .* w(ones(nf,1),:);
end
```

Figura 4.9. Pasos para obtener las ventanas.

```
f=rfft(z.');
[m,a,b]=melbankm(p,n,fs,fl,fh,w);
```

Figura 4.10. Primeros dos pasos en la obtención de los coeficientes.

```
function [x,mn,mx]=melbankm(p,n,fs,fl,fh,w)
f0=700/fs;
fn2=floor(n/2);
lr=log((f0+fh)/(f0+fl))/(p+1);
b1=floor(bl(1))+1;
b2=ceil(bl(2));
b3=floor(bl(3));
  b4=min(fn2,ceil(bl(4)))-1;
  pf=log((f0+(b1:b4)/n)/(f0+fl))/lr;
  fp=floor(pf);
  pm=pf-fp;
  k2=b2-b1+1;
  k3=b3-b1+1;
  k4=b4-b1+1;
  r=[fp(k2:k4) 1+fp(1:k3)];
  c=[k2:k4 1:k3];
  v=2*[1-pm(k2:k4) pm(1:k3)];
  mn=b1+1;
  mx=b4+1;
if nargout > 1
  x=sparse(r,c,v);
else
  x=sparse(r,c+mn-1,v,p,1+fn2);
end
```

Figura 4.11 Cálculo del banco de filtros a aplicar.

El cálculo de los filtros se realiza con la implementación de la ecuación

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

El código con el desarrollo de la ecuación para el cálculo se muestra a continuación en la figura 4.11.

La gráfica de cómo queda el banco de filtros que se va a aplicar a las ventanas se ve en la figura 4.12, donde en eje de las abscisas representa la frecuencia y el eje de las ordenadas la amplitud. Como se puede observar los primeros filtros se encuentran espaciados linealmente, pero a medida de que las frecuencias aumentan, los espacios entre los filtros se vuelven cada vez más grandes.

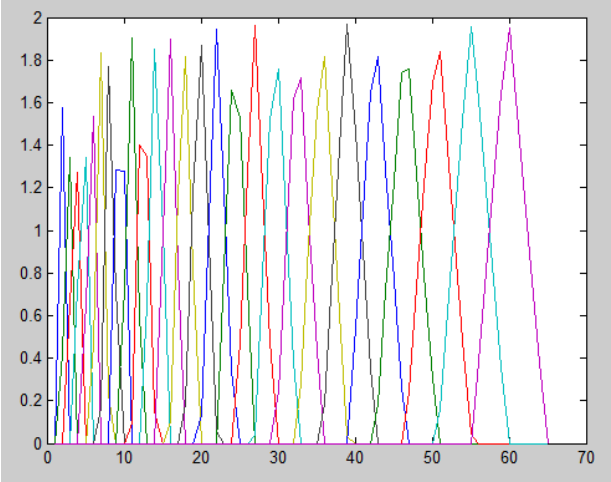

Figura 4.12. Banco de filtros.

En la figura 4.13 se muestra el código utilizado para aplicar los filtros en cada una de las ventanas obtenidas previamente.

```
pw=f(a:b,:).*conj(f(a:b,:));
pth=max(pw(:))*1E-6;
ath=sqrt(pth);
y=log(max(m*abs(f(a:b,:)),ath));
```

Figura 4.13. Aplicación de los filtros a las ventanas.

Por último, se aplica la transformada de coseno discreta con la función *rdct,* como en la figura 4.14. Una vez aplicada esta función, el algoritmo llega a su fin y con esto obtenemos los coeficientes MFCC que representan la envolvente espectral de la señal de voz proporcionada.

```
c=rdct(y).';
```

Figura 4.14 Cálculo de transformada de coseno discreta.

La transformada de coseno discreto se realiza mediante la ecuación siguiente

$$Y[k] = e^{j\frac{\pi}{2N}k} C_x[k]$$

El código de la figura 4.15 muestra la implementación de la fórmula anterior.

```
function y=rdct(x,n,a,b)
fl=size(x,1)==1;
if fl x=x(:); end
[m,k]=size(x);
if nargin<4 b=1;
    if nargin<3 a=sqrt(2*m);
        if nargin<2 n=m;
        end
    end
end
if n>m x=[x; zeros(n-m,k)];
elseif n<m x(n+1:m,:)=[];
end
x=[x(1:2:n,:); x(2*fix(n/2):-2:2,:)];
z=[sqrt(2) 2*exp((-0.5i*pi/n)*(1:n-1))].';
y=real(fft(x).*z(:,ones(1,k)))/a;
y(1,:)=y(1,:)*b;
if fl y=y.';

end
```

Figura 4.15. Código de la implementación de la fórmula.

4.2 Función gmm_estimate

Esta función permite el cálculo de las gaussianas para los datos de entrenamiento, recibe tres parámetros que son los coeficientes MFCC obtenidos previamente, el número de gaussianas definidas al inicio y el número de iteraciones, como se muestra en la figura 4.16. Los valores que se obtendrán son aquellos que servirán para representar matemáticamente los modelos de los hablantes, estos parámetros son las medias, varianzas y los pesos correspondientes a cada componente gaussiana.

```
[mu_train1,sigma_train1,c_train1]=gmm_estimate(training_features1',
No_of_Gaussians,1);
disp('Completed Training Speaker 1 model (Press any key to
continue)');
pause;

[mu_train2,sigma_train2,c_train2]=gmm_estimate(training_features2',
No_of_Gaussians,1);
disp('Completed Training Speaker 2 model (Press any key to
continue)');
pause;
```

Figura 4.16. Llamada a la función gmm_estimate.

Se comienza por definir algunas variables a utilizar durante el proceso, figura 4.17. Una vez definidas estas variables se procede con el proceso iterativo para el algoritmo expectation-maximization, figura 4.18.

```
  % GENERAL PARAMETERS
  [L,T]=size(X);          % data length
  varL=var(X')';     % variance for each row data;
  min_diff_LLH=0.001;    % convergence criteria
  % DEFAULTS
  if nargin<3  iT=10; end    % number of iterations, by defaut 10
  if nargin<4  mu=X(:,[fix((T-1).*rand(1,M))+1]); end % mu def: M
rand vect.
  if nargin<5  sigm=repmat(varL./(M.^2),[1,M]); end % sigm def:
same variance
  if nargin<6  c=ones(M,1)./M; end  % c def: same weight
  if nargin<7  Vm=4; end    % minimum variance factor
  min_sigm=repmat(varL./(Vm.^2*M.^2),[1,M]);    % MINIMUM sigma!
  if DEBUG sqrt(devs),sqrt(sigm),pause;end
  % VARIABLES
  lgam_m=zeros(T,M);     % prob of each (X:,t) to belong to the kth
mixture
  lB=zeros(T,1);          % log-likelihood
  lBM=zeros(T,M);         % log-likelihhod for separate mixtures
  old_LLH=-9e99;          % initial log-likelihood
```

Figura 4.17. Variables del algoritmo expectation-maximization.

```
% START ITERATATIONS
  for iter=1:iT
    if GRAPH graph_gmm(X,mu,sigm,c),pause,end
    if DEBUG disp(['*********** ',num2str(iter),' ************']);
```

Figura 4.18. Inicio de las iteraciones.

Para el paso expectation debemos calcular la siguiente probabilidad

$$P(q_k|x_n,\theta) = \frac{P(q_k|\theta)\cdot P(x_n|q_k,\theta)}{P(x_n|\theta)}$$

En la figura 4.19 se muestra el código de este paso, donde

$$c(k) = P(q_k|\theta)$$

$$lBM(n,k) = \log p(x_n|q_k,\theta)$$

$$lB(k) = \log p(x_n|\theta)$$

$$gam\_m(n,k) = P(q_k|x_n,\theta)$$

Para calcular los nuevos pesos se usa la siguiente fórmula, la implementación está en la figura 4.20.

$$P\left(q_k{}^{(new)}\middle|\theta^{(new)}\right) = \frac{1}{T}\sum_{n=1}^{T} P(q_k|x_n,\theta)$$

```
    % EXPECTATION STEP
***************************************************
    [lBM,lB]=lmultigauss(X,mu,sigm,c);
    %%Y, YM]
    %lBMA,lBA]
    if DEBUG lB,B=exp(lB),pause; end
    LLH=mean(lB);
    disp(sprintf('log-likelihood :  %f',LLH));

    lgam_m=lBM-repmat(lB,[1,M]);  % logarithmic version
    gam_m=exp(lgam_m);            % linear version

    % MAXIMIZATION STEP
***************************************************
    sgam_m=sum(gam_m);            % sum of gam_m for all X(:,t)
```

Figura 4.19 Implementación del algoritmo expectation-maximization.

```
    % gaussian weights ********************************
    new_c=mean(gam_m)';
```

Figura 4.20. Obtención de los pesos.

Para calcular las medias se tiene la siguiente fórmula y el desarrollo del código se muestra en la figura 4.21.

$$\mu_k{}^{(new)} = \frac{\sum_{n=1}^{T} x_n P\left(q_k \middle| x_n, \theta\right)}{\sum_{n=1}^{T} P\left(q_k \middle| x_n, \theta\right)}$$

```
% means     ****************************************
% (convert gam_m and X to (L,M,T) and .* and then sum over T)
mu_numerator=sum(permute(repmat(gam_m,[1,1,L]),[3,2,1]).*...
           permute(repmat(X,[1,1,M]),[1,3,2]),3);
% convert  sgam_m(1,M,N) -> (L,M,N) and then ./
new_mu=mu_numerator./repmat(sgam_m,[L,1]);
```

Figura 4.21 Actualización de las medias.

Para calcular las varianzas

$$\Sigma_k{}^{(new)} = \frac{\sum_{n=1}^{T} P\left(q_k \middle| x_n, \theta\right)\left(x_n - \mu_k{}^{(new)}\right)\left(x_n - \mu_k{}^{(new)}\right)^T}{\sum_{n=1}^{T} P\left(q_k \middle| x_n, \theta\right)}$$

En la figura 4.22 está el desarrollo de la fórmula anterior y la actualización de los parámetros, donde

$$new\_mu(:,k) = \mu_k{}^{(new)}$$

$$new\_sigm(:,k) = \Sigma_k{}^{(new)}$$

$$new\_c(k) = P\left(q_k{}^{(new)} \middle| \theta^{(new)}\right)$$

```
% variances ****************************************
sig_numerator=sum(permute(repmat(gam_m,[1,1,L]),[3,2,1]).*...
            permute(repmat(X.*X,[1,1,M]),[1,3,2]),3);
new_sigm=sig_numerator./repmat(sgam_m,[L,1])-new_mu.^2;
% the variance is limited to a minimum
new_sigm=max(new_sigm,min_sigm);

%*******% UPDATE
if old_LLH>=LLH-min_diff_LLH
  disp('converge');
  break;
else
    old_LLH=LLH;
  mu=new_mu;
  sigm=new_sigm;
  c=new_c;
end
```

Figura 4.22. Actualización de parámetros.

Las gráficas obtenidas del algoritmo GMM muestran en el eje vertical el valor de densidad que se obtuvo por cada valor de los coeficientes MFCC, mientras que en el eje horizontal se encuentran los

valores de los coeficientes. Para el primer hablante se obtiene la gráfica de la figura 4.23, en la cual al fondo se observa la energía de la señal. Las gaussianas representadas con las líneas delgadas fue lo que se obtuvo con la función *gmm_estimate* y por último la que tiene la línea más gruesa es la suma total de las anteriores. Del segundo hablante se obtiene la gráfica de la figura 4.24, al igual que con la figura 4.23 se tienen la energía, las gaussianas calculadas y la suma de las mismas.

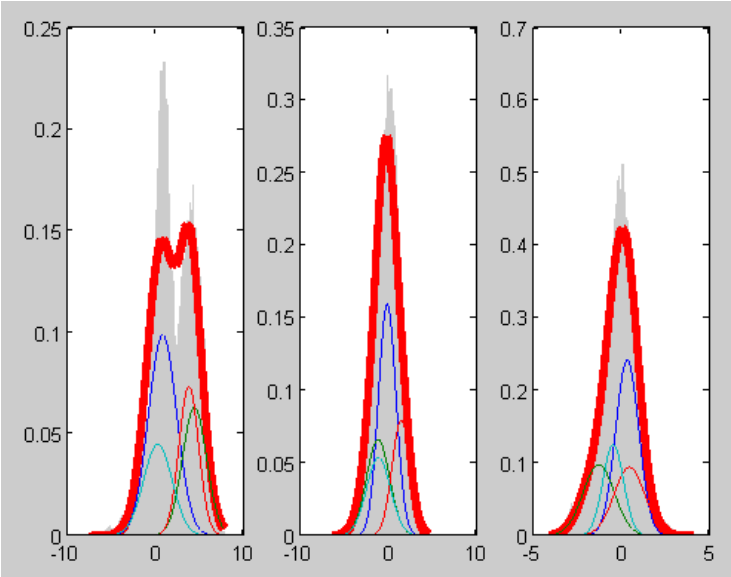

Figura 4.23. Coeficientes para el primer modelo.

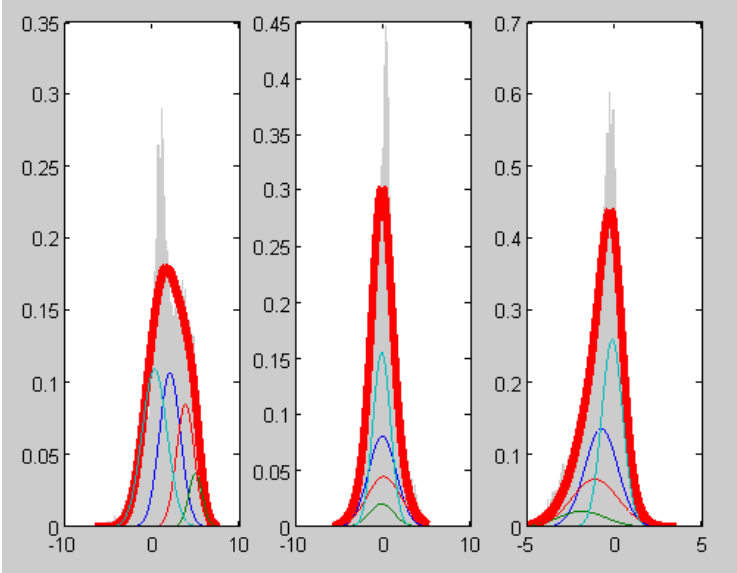

Figura 4.24. Coeficientes para el segundo modelo.

Capítulo 5. Experimentos

Gracias a MFCC logramos obtener información que puede discriminar bien las características de las voces de cada uno de los individuos; éste es un elemento de suma importancia, ya que sin los coeficientes obtenidos mediante este algoritmo podríamos tener datos que no representen correctamente de forma matemática las voces.

Por último, se encuentra el algoritmo GMM, el que hace un excelente trabajo al momento de obtener los modelos representativos de las voces de cada persona. Estos modelos creados a partir de los coeficientes MFCC logran hacer que el reconocimiento sea preciso y se pueda distinguir la voz de una persona en particular. Para las pruebas se tomaron 140 archivos de audio para los hombres y 130 para mujeres, se dividieron en diferentes proporciones para los datos utilizados en el entrenamiento y para las pruebas.

Durante las pruebas la cantidad de parámetros utilizados son 14 coeficientes para la extracción de características de las voces de los hablantes y 10 componentes gaussianas para cada uno de los modelos.

En el primer experimento se utilizaron el 50% de los datos para la etapa de entrenamiento y en la de prueba por igual. Los resultados obtenidos se presentan en la siguiente tabla 5.1 con la cantidad de personas identificadas correcta e incorrectamente.

50 % entrenamiento 50% de prueba

	Correctos	Incorrectos	Porcentaje de exactitud
Hombres	136	4	97.1429
Mujeres	126	4	96.9231

Tabla 5.1. 50 % entrenamiento.

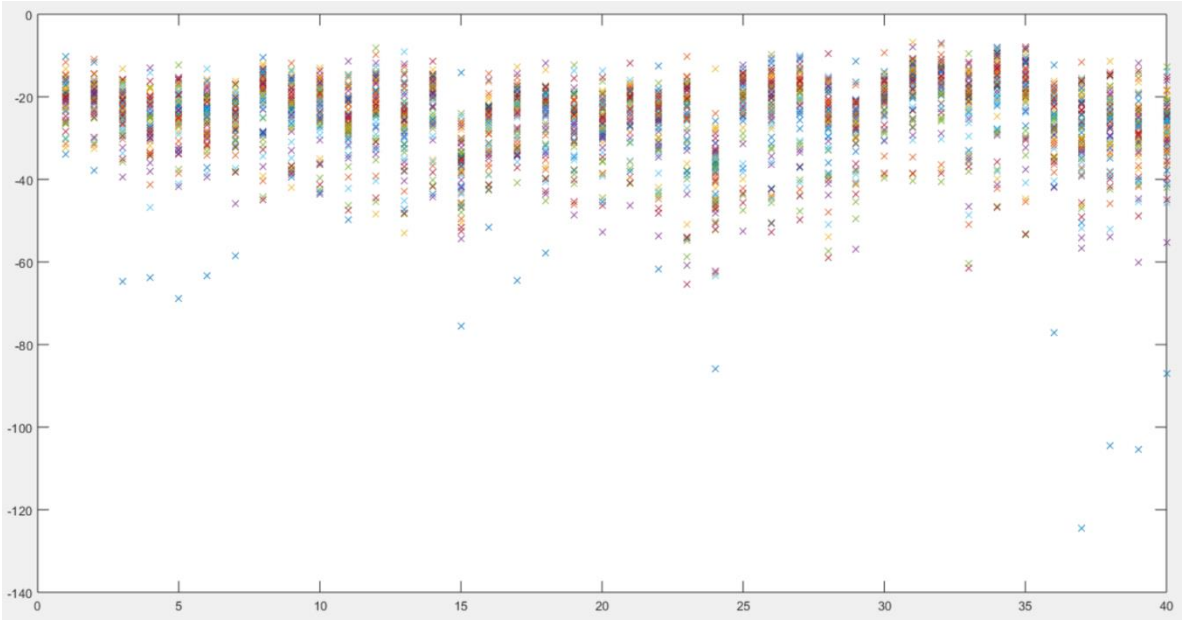

Figura 5.1. Resultados de los hombres con un 50% de datos para entrenamiento.

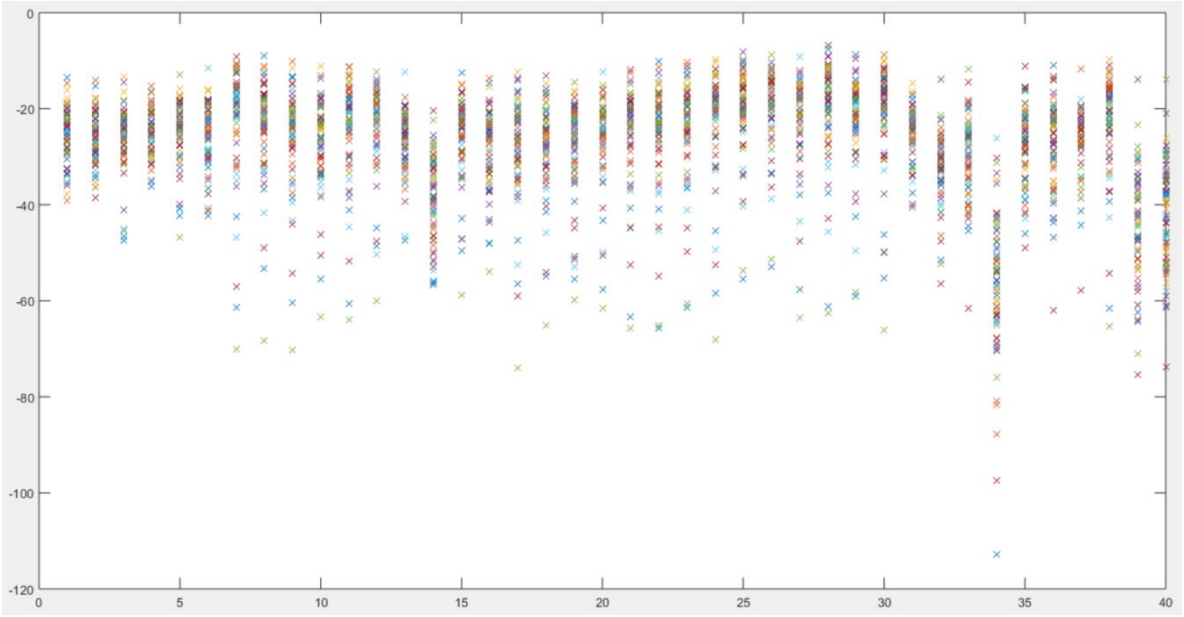

Figura 5.2. Resultados de las mujeres con un 50% de datos para entrenamiento.

La figura 5.1 y 5.2 nos muestra los valores obtenidos después de haberse realizado todo el algoritmo; en el eje de las ordenadas se tiene el log-likelihood y en el de las abscisas los hablantes, donde cada x representa el valor obtenido al realizar la comparación con el resto de los modelos. Las gráficas sirven para ilustrar qué tan buena es la discriminación entre el hablante identificado y el resto.

El valor más cercano al cero corresponde el hablante que el sistema determinó con la mayor probabilidad. Se puede observar que en algunos casos se tienen valores muy cercanos, esto significa que se encontró personas con voces muy similares y hay otros dónde los valores están más separados, indicando que en esas voces existe menos similitud con el resto.

En el segundo experimento se disminuyó la cantidad de elementos utilizados para la etapa de entrenamiento al 40% y el 60 para la etapa de prueba con los siguientes resultados.

40 % entrenamiento 60% de prueba

	Correctos	Incorrectos	Porcentaje de exactitud
Hombres	137	3	97.8571
Mujeres	128	2	98.4615

Tabla 5.2. 40 % entrenamiento.

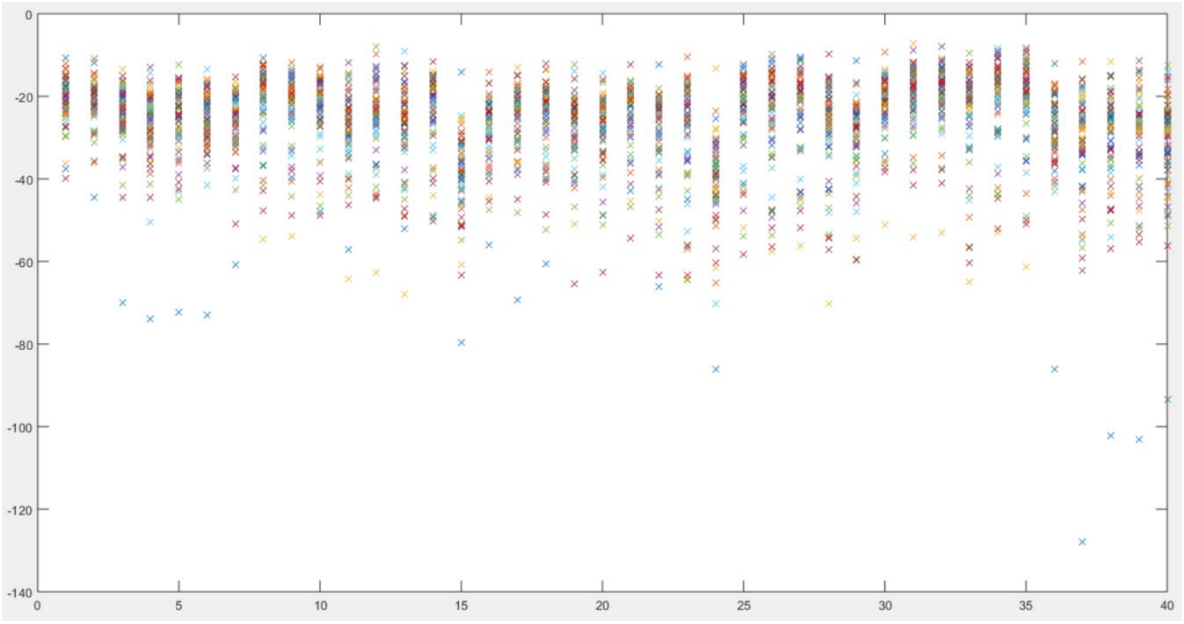

Figura 5.3. Resultados de los hombres con un 40% de datos para entrenamiento.

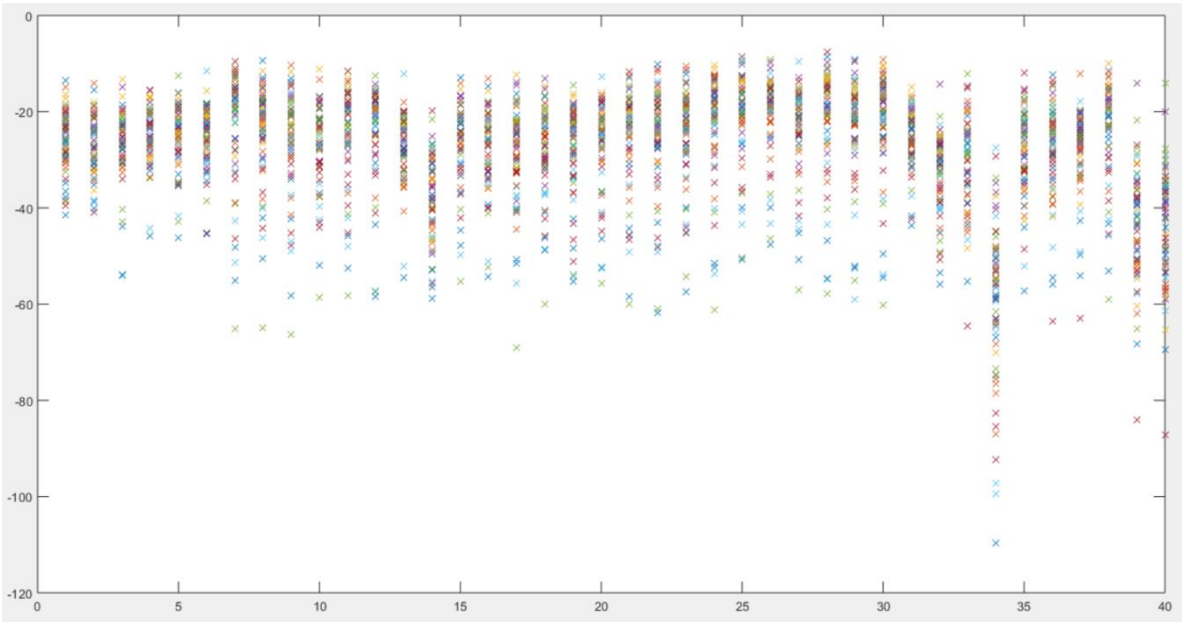

Figura 5.4. Resultados de las mujeres con un 40% de datos para entrenamiento.

El tercer experimento consistió en reducir una vez más la cantidad de datos en el entrenamiento, esta vez a un 35%.

35 % entrenamiento 65% de prueba

	Correctos	Incorrectos	Porcentaje de exactitud
Hombres	136	4	97.1429
Mujeres	129	1	99.2308

Tabla 5.3. 35 % entrenamiento.

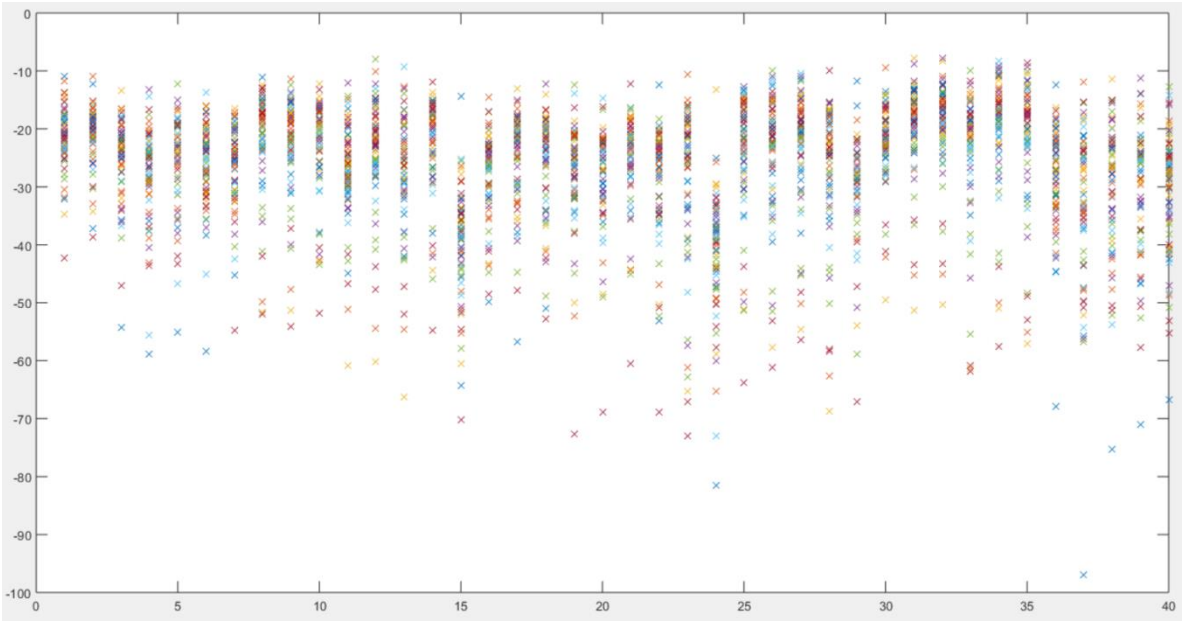

Figura 5.5. Resultados de los hombres con un 35% de datos para entrenamiento.

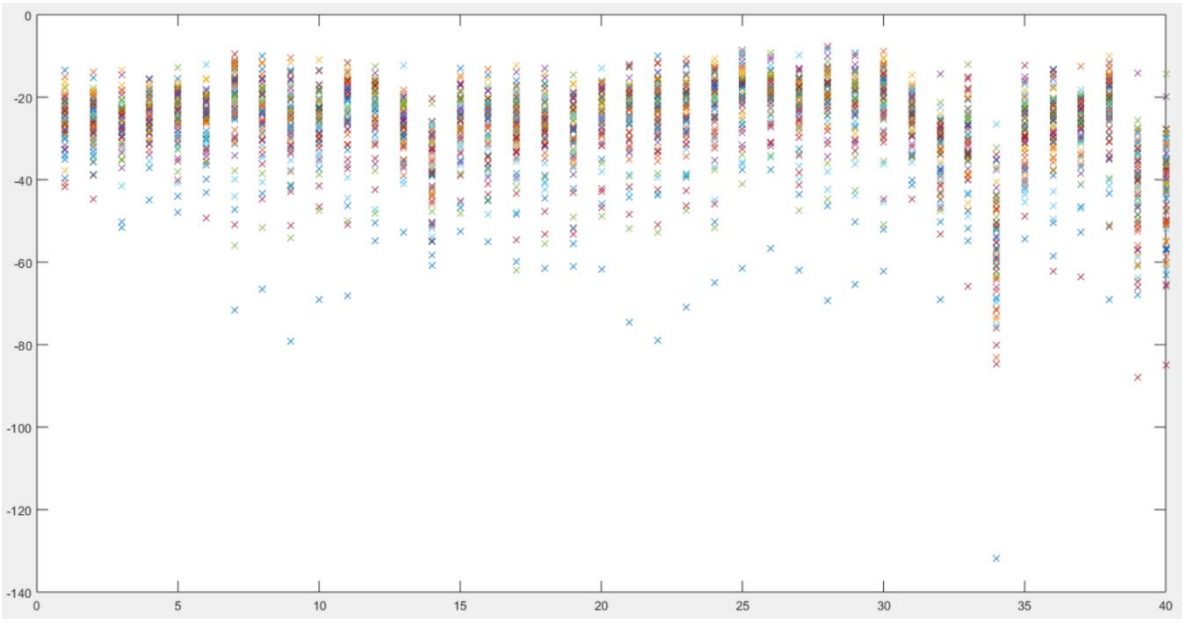

Figura 5.6. Resultados de las mujeres con un 35% de datos para entrenamiento.

Para el siguiente se disminuyó el porcentaje para el entrenamiento a 30% y el resto para prueba.

30 % entrenamiento 70% de prueba

	Correctos	Incorrectos	Porcentaje de exactitud
Hombres	134	6	95.7143
Mujeres	126	4	96.9231

Tabla 5.4. 30 % entrenamiento

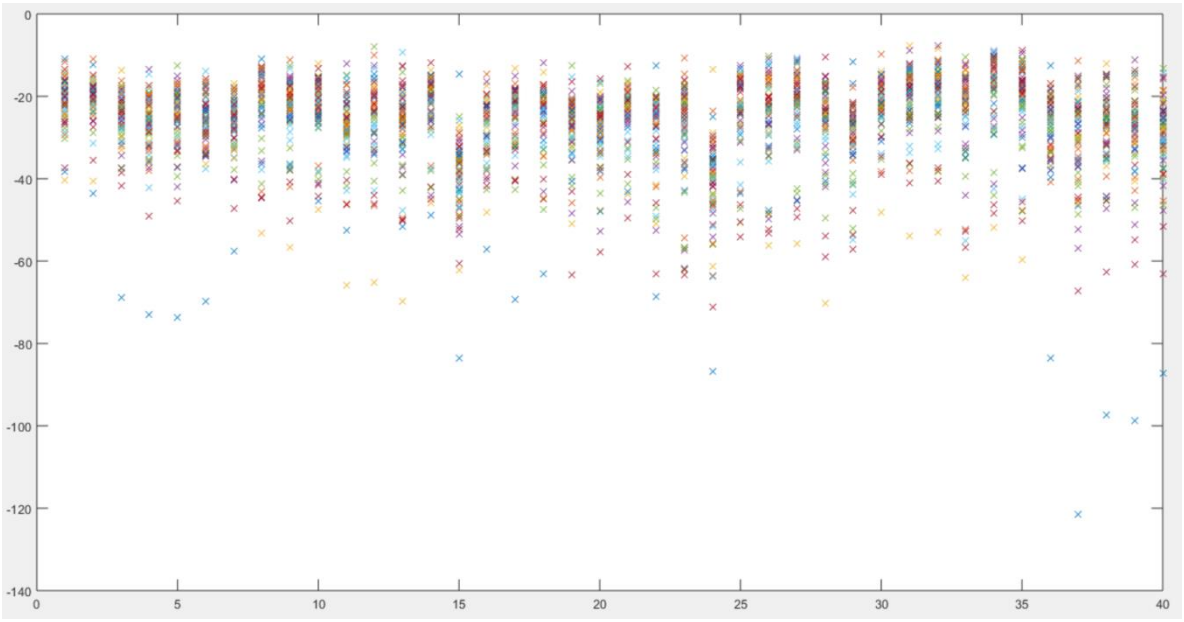

Figura 5.7. Resultados de los hombres con un 30% de datos para entrenamiento.

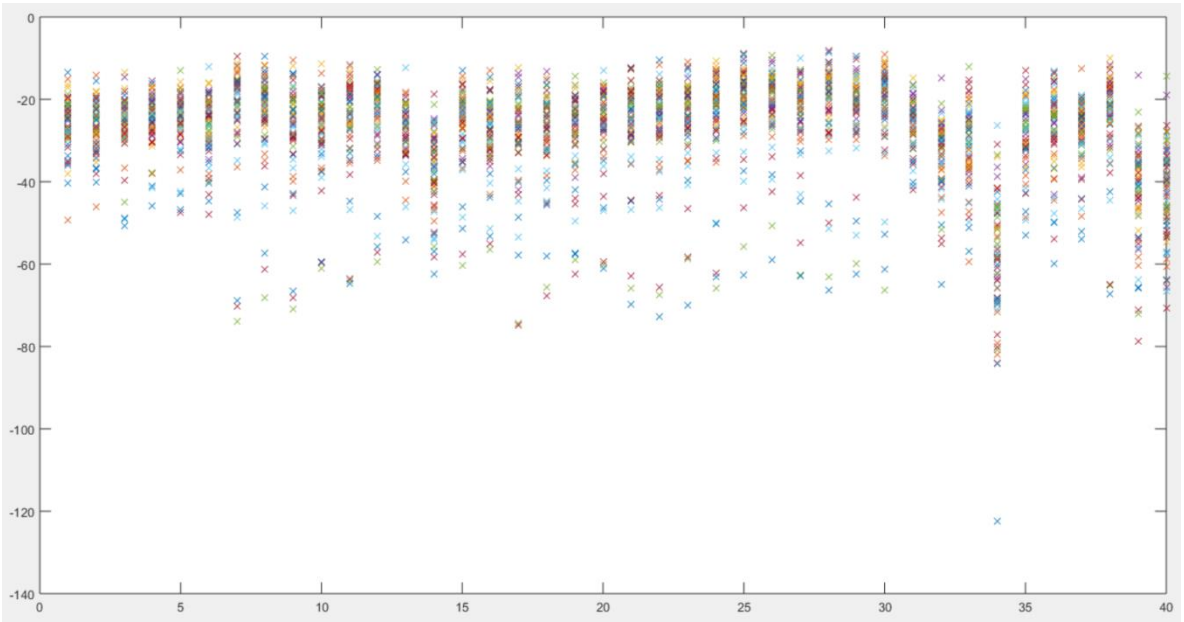

Figura 5.8. Resultados de las mujeres con un 30% de datos para entrenamiento.

Posteriormente se hizo otro decremento de 5% en lo datos de entrenamiento, quedando con un total de 25%.

25 % entrenamiento 75% de prueba

	Correctos	Incorrectos	Porcentaje de exactitud
Hombres	135	5	96.4286
Mujeres	126	4	96.9231

Tabla 5.5. 25 % entrenamiento

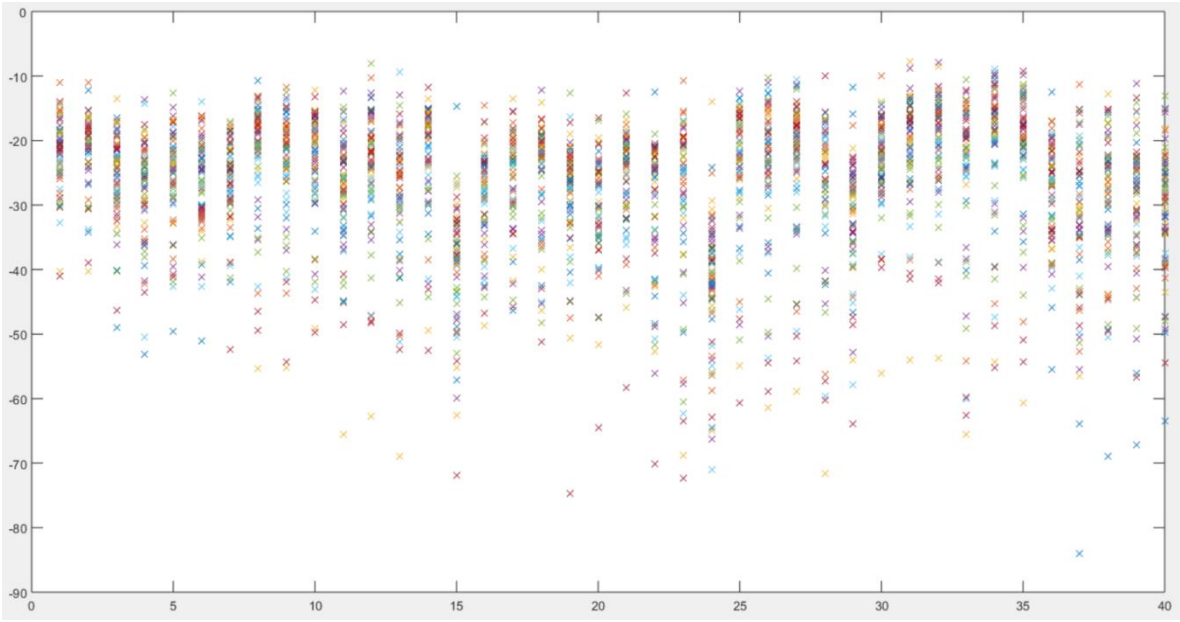

Figura 5.9. Resultados de los hombres con un 25% de datos para entrenamiento.

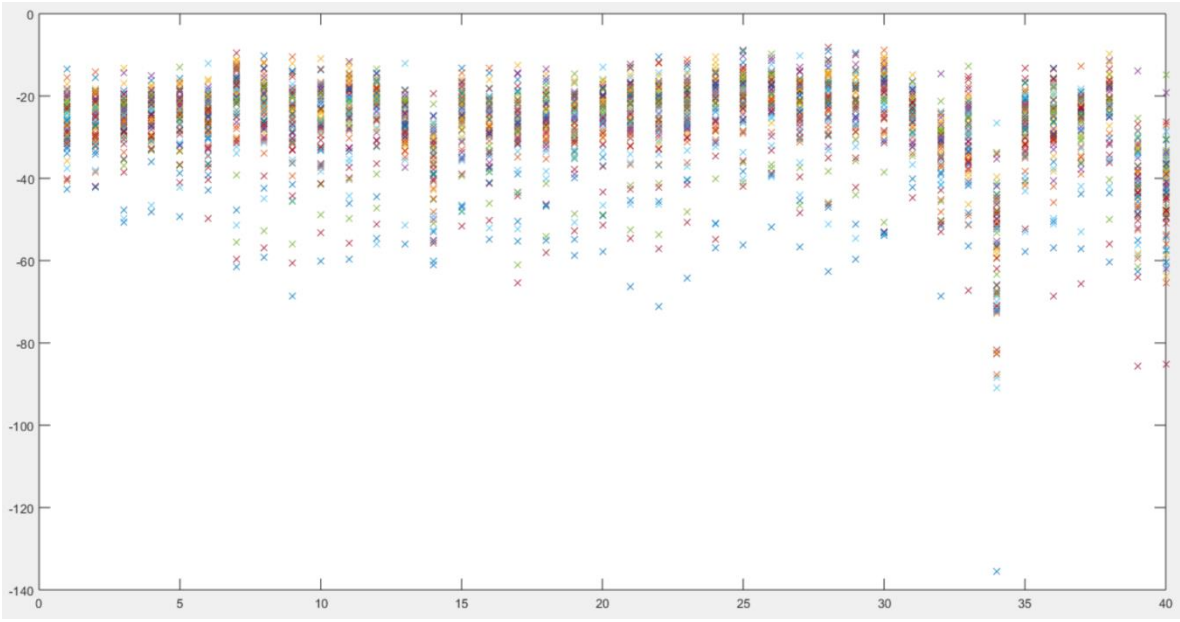

Figura 5.10. Resultados de las mujeres con un 25% de datos para entrenamiento.

Para el último experimento se tomó únicamente el 20% de datos para el entrenamiento y 80% para la prueba obteniendo lo siguiente.

20 % entrenamiento 80% de prueba

	Correctos	Incorrectos	Porcentaje de exactitud
Hombres	131	9	93.5714
Mujeres	124	6	95.3846

Tabla 5.6. 20 % entrenamiento.

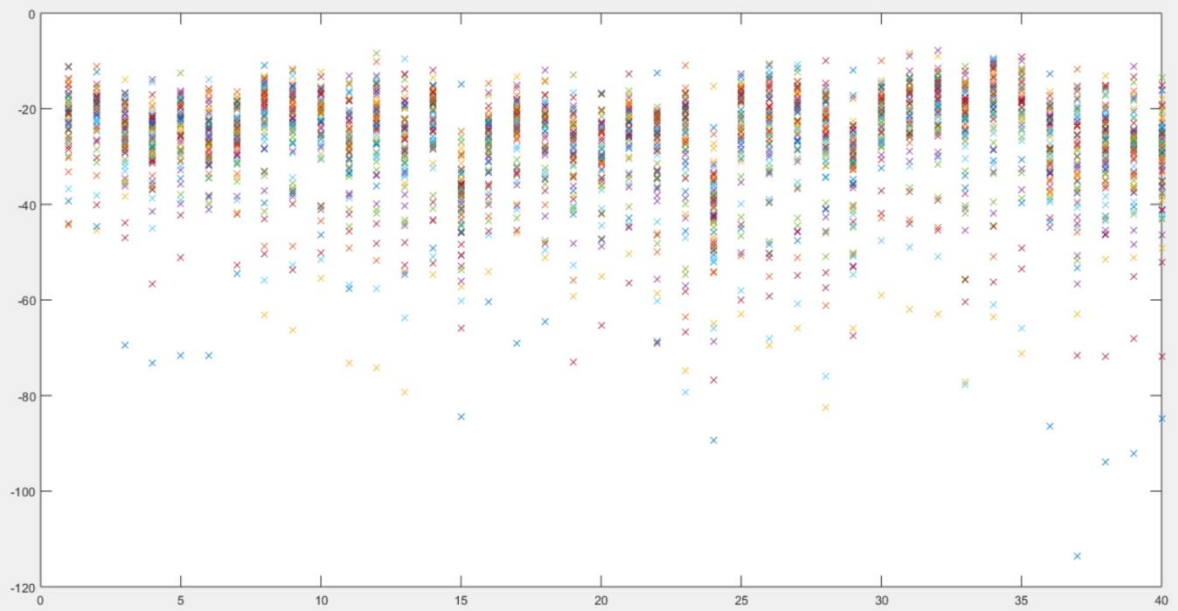

Figura 5.11. Resultados de los hombres con un 20% de datos para entrenamiento.

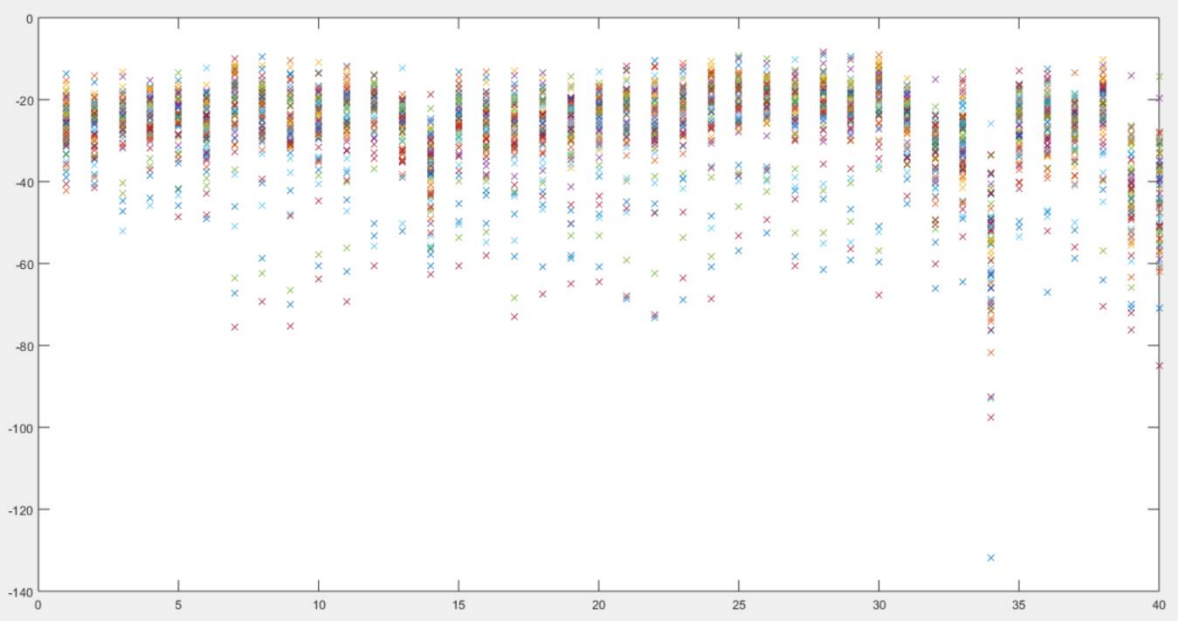

Figura 5.12. Resultados de las mujeres con un 20% de datos para entrenamiento.

Como se puede apreciar en los resultados obtenidos, aunque el porcentaje de datos utilizados sea menor para realizar el modelo del hablante, el algoritmo es capaz de crear una representación lo suficientemente buena para poder discriminar entre las voces de las diferentes personas que se encuentran en el corpus. Esto se debe a los métodos utilizados durante el proceso de reconocimiento, desde la extracción de las características de la voz haciendo uso de MFCC y posteriormente utilizando un buen algoritmo de reconocimiento de patrones como lo es GMM.

5.1 Experimentos combinando hombres y mujeres

Las siguientes pruebas se realizaron mezclando voces de hombres y mujeres en los archivos de entrenamiento y prueba, con el fin de determinar si el sistema es capaz de realizar la identificación correcta de los individuos, aun siendo de ambos géneros. La meta es saber si se presenta algún error, en el que se haya identificado una persona de sexo opuesto al de la persona a partir de la que se generó el modelo. Los datos utilizados son un total de 80 personas, las cuales se encuentran divididas en 40 mujeres y 40 hombres; se tomaron grabaciones de todas las categorías, incluidas rango de edades y tipo de llamada. Una vez realizadas estas pruebas, también se podrá determinar si el ruido ambiental, incluido en algunas de las grabaciones, especialmente en aquellas que se hicieron en plena vía pública, es lo suficientemente robusto cómo para producir errores significativos en el sistema de identificación.

El primer experimento se realizó usando el 50% de datos para el entrenamiento y el otro 50% se utilizó para poder compararlos contra los modelos generados, de esta forma arrojando el resultado de la identificación.

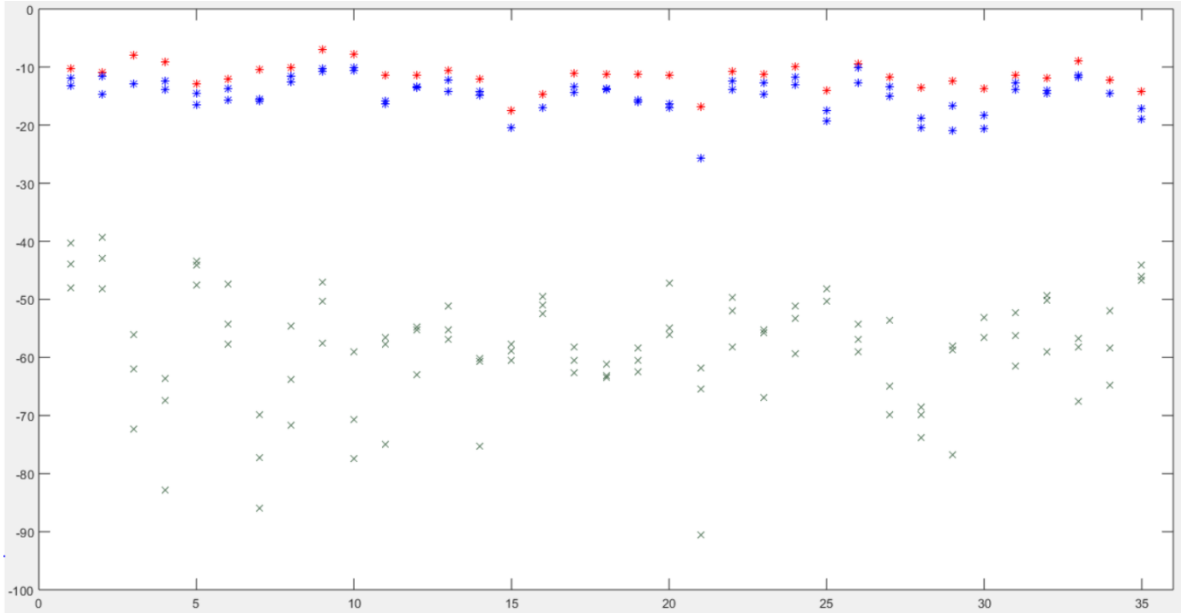

Figura 5.13. Resultados con un 50% de datos para entrenamiento – parte 1.

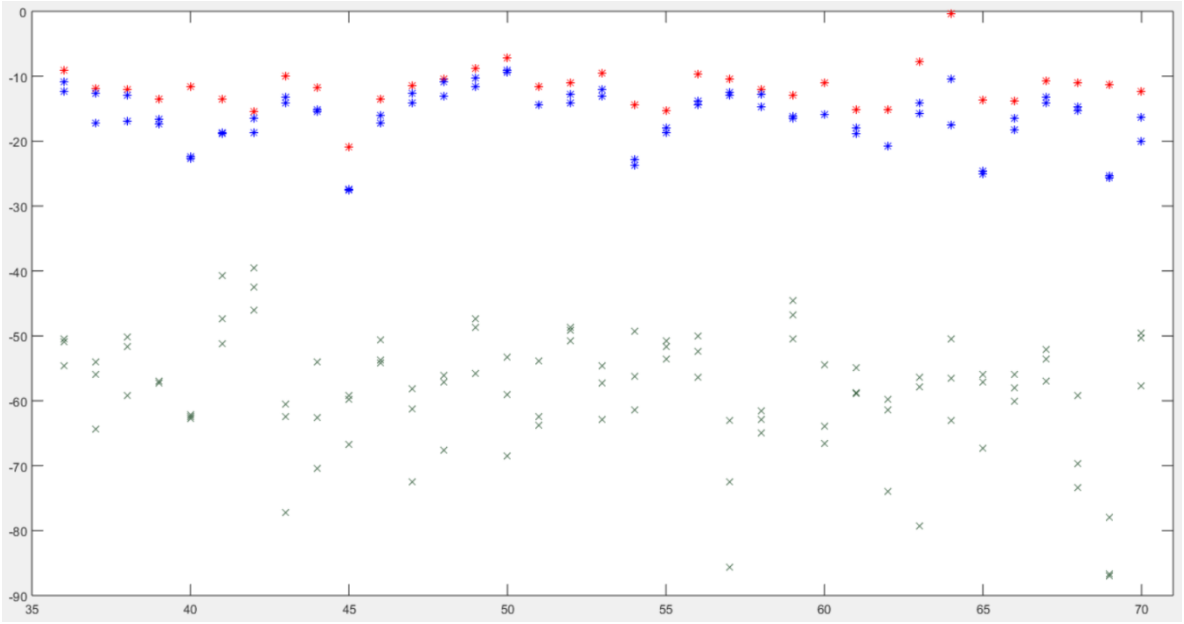

Figura 5.14. Resultados con un 50% de datos para entrenamiento – parte 2.

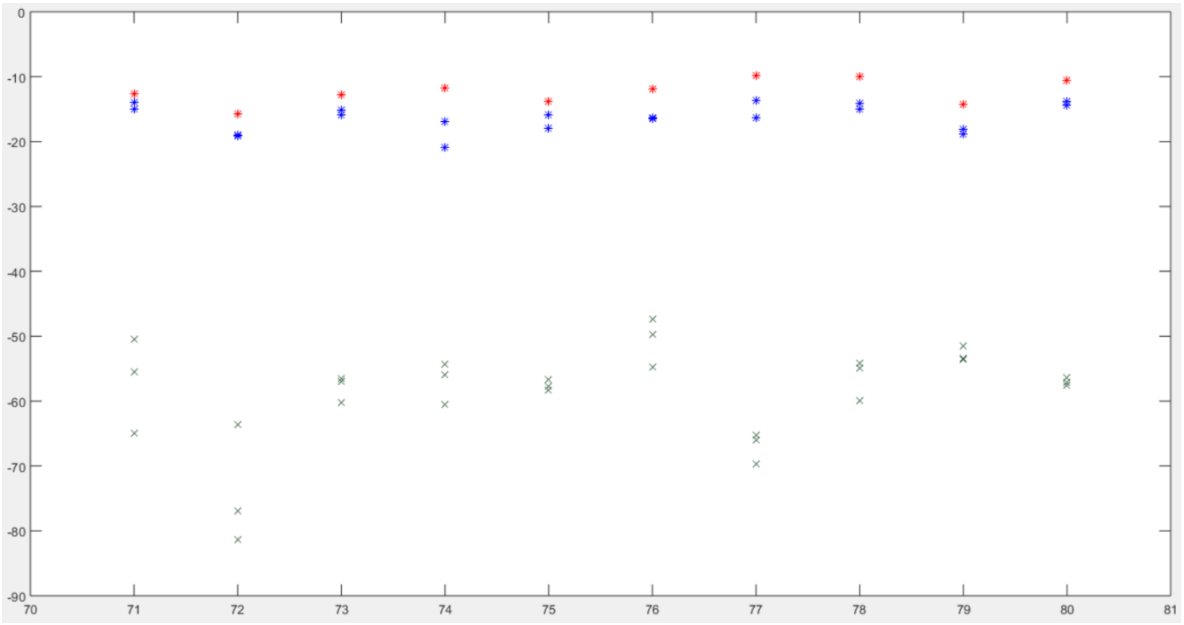

Figura 5.15. Resultados con un 50% de datos para entrenamiento – parte 3.

En la tabla 5.7 se muestra el error arrojado por el sistema; en este caso la persona real era una mujer perteneciente al rango de edades de 18 a 30 y la que se identificó también fue una mujer correspondiente al mismo rango de edades. Como se puede observar, el error del sistema no fue tan grave, ya que fueron personas del mismo género y rango de edad e incluso el mismo tipo de llamada que fue teléfono público a celular.

Rango de edades de los hablantes y género	
Real	Identificado
Mujer 02-a-01-0004srv.wav 18-30	Mujer 02-a-01-0001srv.wav 18-30

Tabla 5.7. Resultados de la identificación, para 50% de datos para entrenamiento.

Segundo experimento con el 40% de datos para el entrenamiento y el 60% restante para las pruebas. En las gráficas siguientes, se puede observar que no hay muchos valores muy cercanos entre el primer valor y el segundo, esto quiere decir que hubo una mejor discriminación entre los hablantes. Los resultados se deben a la creación de un buen modelo, lo que nos podría indicar que, a pesar de disminuir la información usada, la porción extraída contiene la suficiente cantidad de información sobre la voz de cada uno de los individuos.

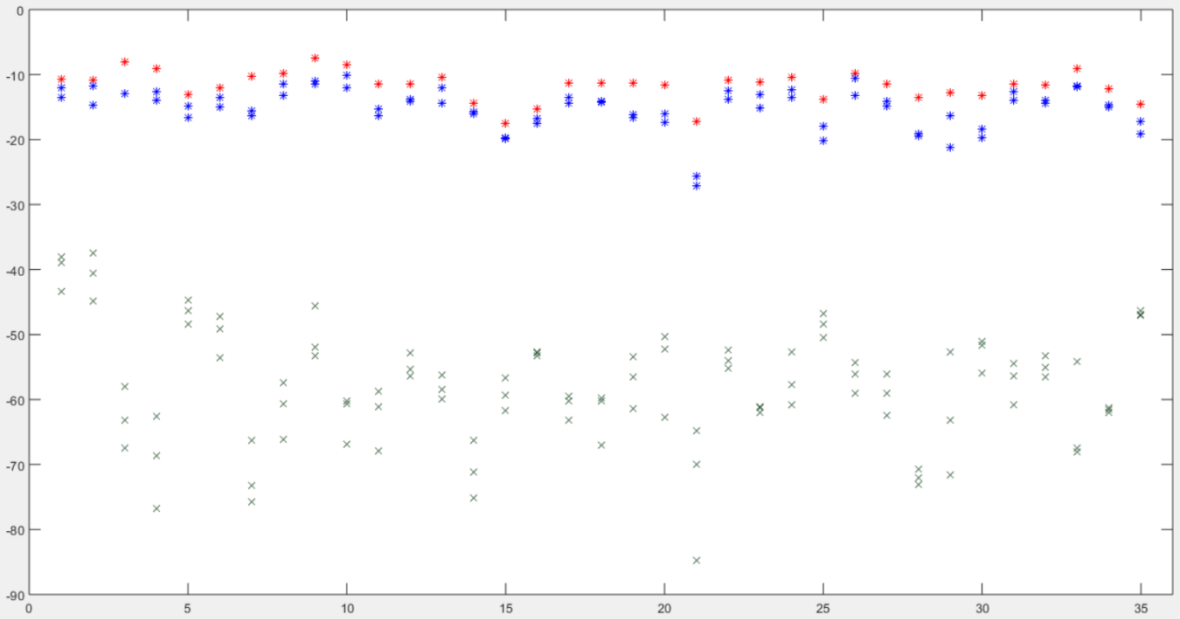

Figura 5.16. Resultados con un 40% de datos para entrenamiento – parte 1.

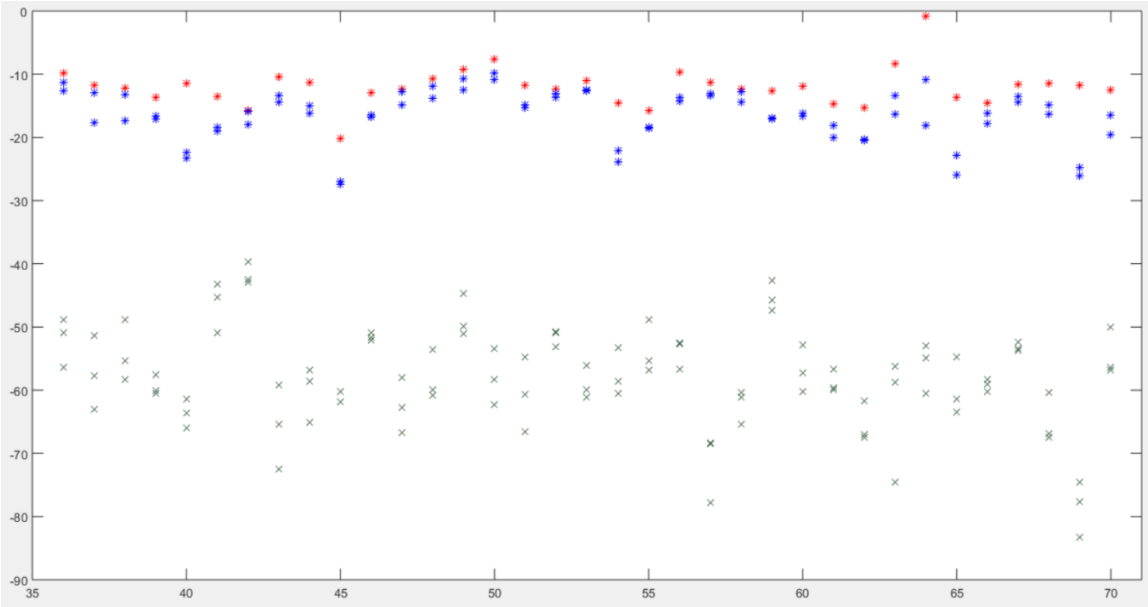

Figura 5.17. Resultados con un 40% de datos para entrenamiento – parte 2.

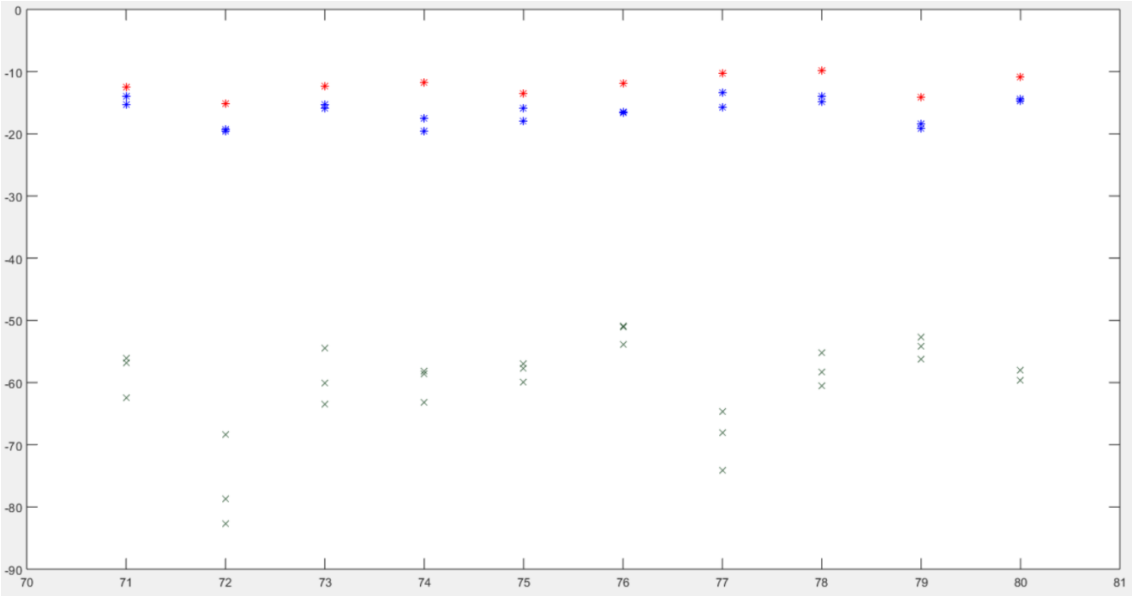

Figura 5.18. Resultados con un 40% de datos para entrenamiento – parte 3.

En el segundo experimento se presentó un error en la identificación, table 5.8. Al igual que en el experimento anterior, tanto la mujer que se identificó, como la verdadera estaban en el rango de 18 a 30.

Rango de edades de los hablantes y género	
Real	Identificado
Mujer 02-a-01-0004srv.wav 18-30	Mujer 02-a-01-0001srv.wav 18-30

Tabla 5.8. Resultados de la identificación, para 40% de datos para entrenamiento.

Tercer experimento con 30% de los datos

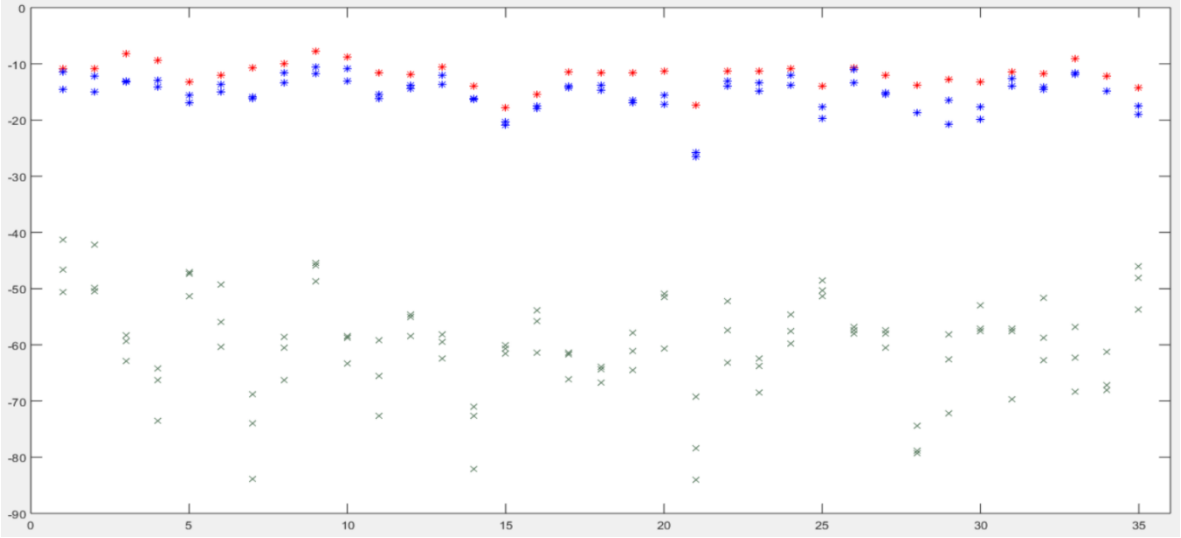

Figura 5.19. Resultados con un 30% de datos para entrenamiento – parte 1.

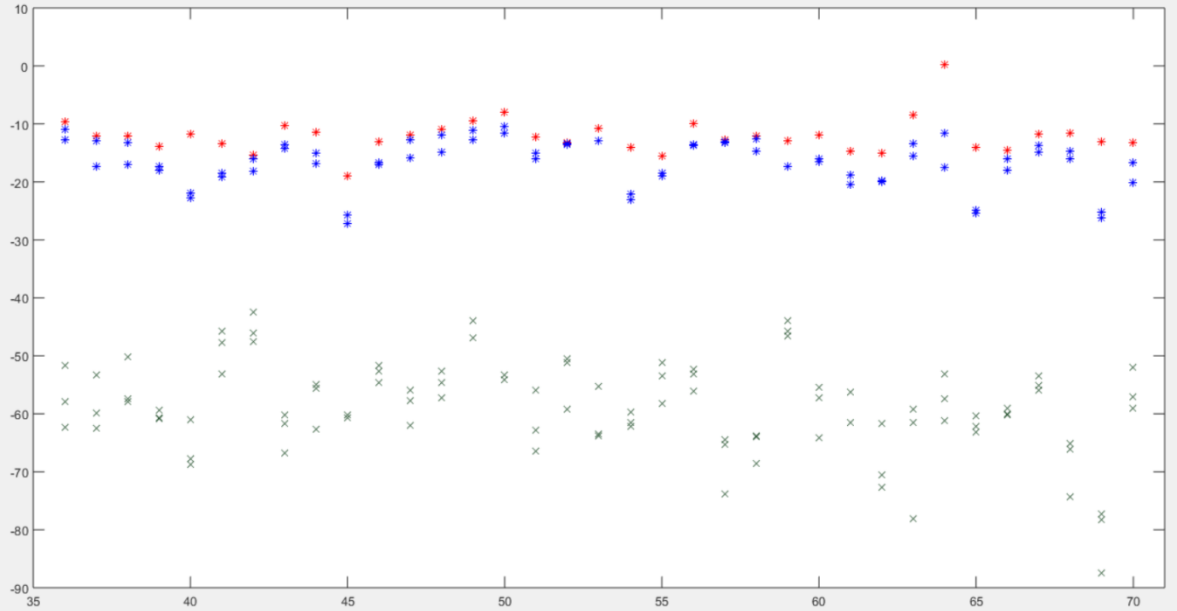

Figura 5.20. Resultados con un 30% de datos para entrenamiento – parte 2.

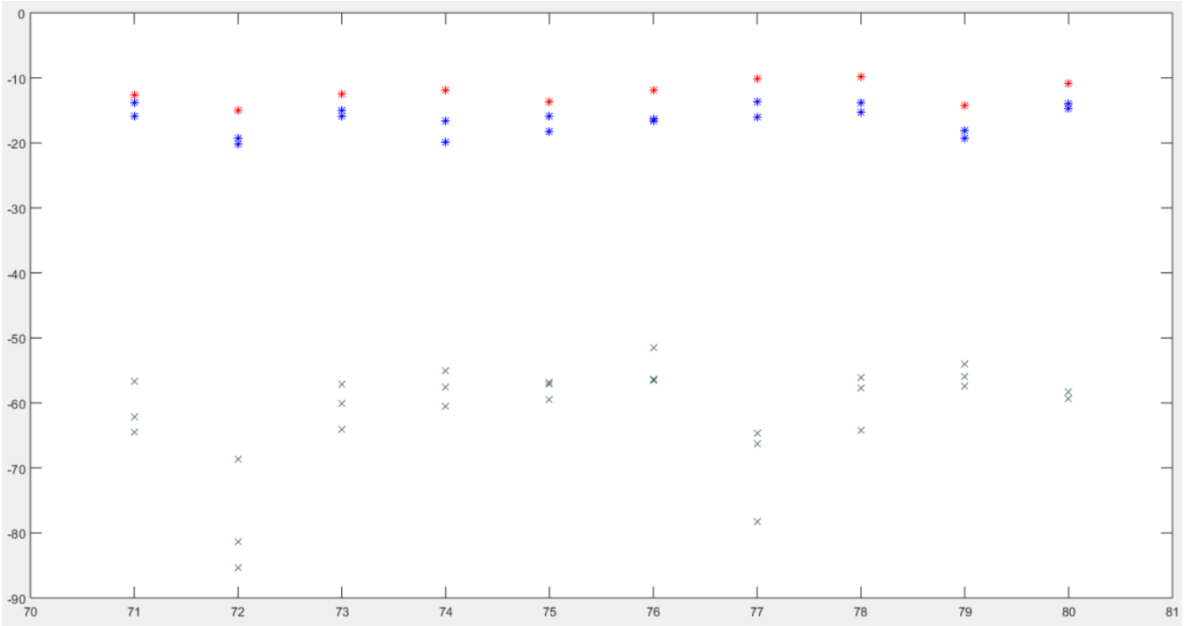

Figura 5.21. Resultados con un 30% de datos para entrenamiento – parte 3.

En el tercer experimento se presentó un error en la identificación. Se muestra que el sistema sigue siendo constante en la identificación, a pesar de contar con una menor cantidad de información para realizar los modelos de los hablantes utilizados, ya que al igual que en los experimentos anteriores la persona identificada erróneamente fue una mujer perteneciente al rango de edades de 18 a 30, que fue identificada como otra mujer que pertenece al mismo rango de edades. A pesar de que se tiene el error, no fue muy grave debido a que ambas fueron mujeres y del mismo rango de edades, esto demuestra cierta constancia en los resultados obtenidos por el sistema.

Rango de edades de los hablantes y género	
Real	Identificado
Mujer 02-b-04-0002srv.wav 31-45	Mujer 02-a-04-0005srv.wav 18-30

Tabla 5.9. Resultados de la identificación, para 30% de datos para entrenamiento.

El cuarto y último experimento se realizó con el 20% de información para el entrenamiento.

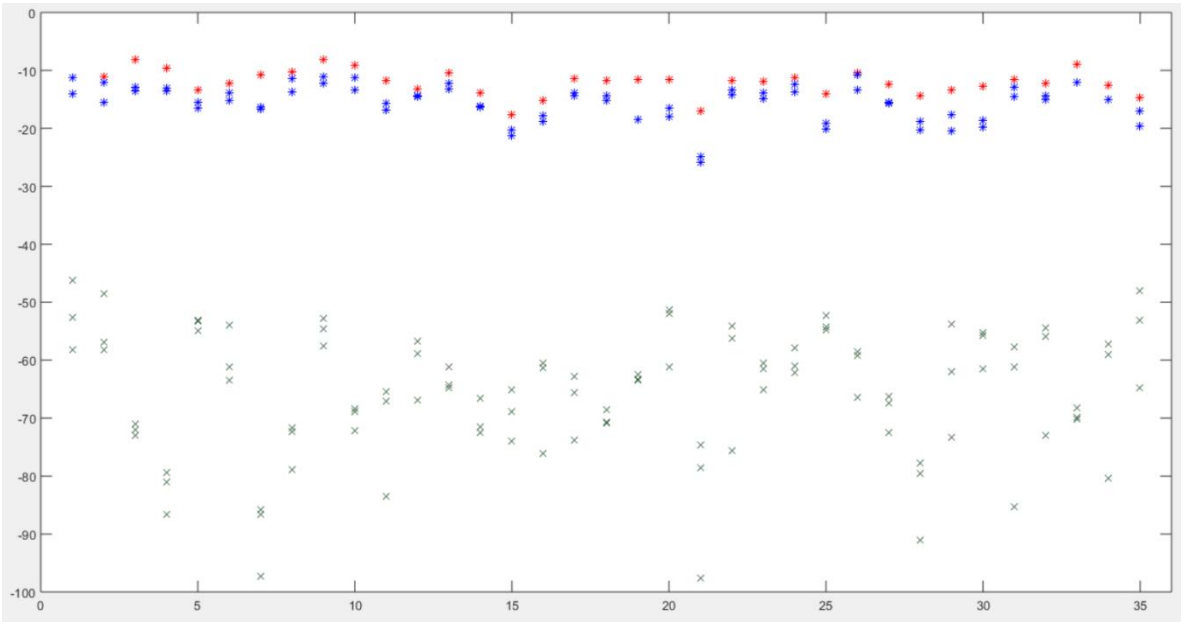

Figura 5.22. Resultados con un 20% de datos para entrenamiento – parte 1.

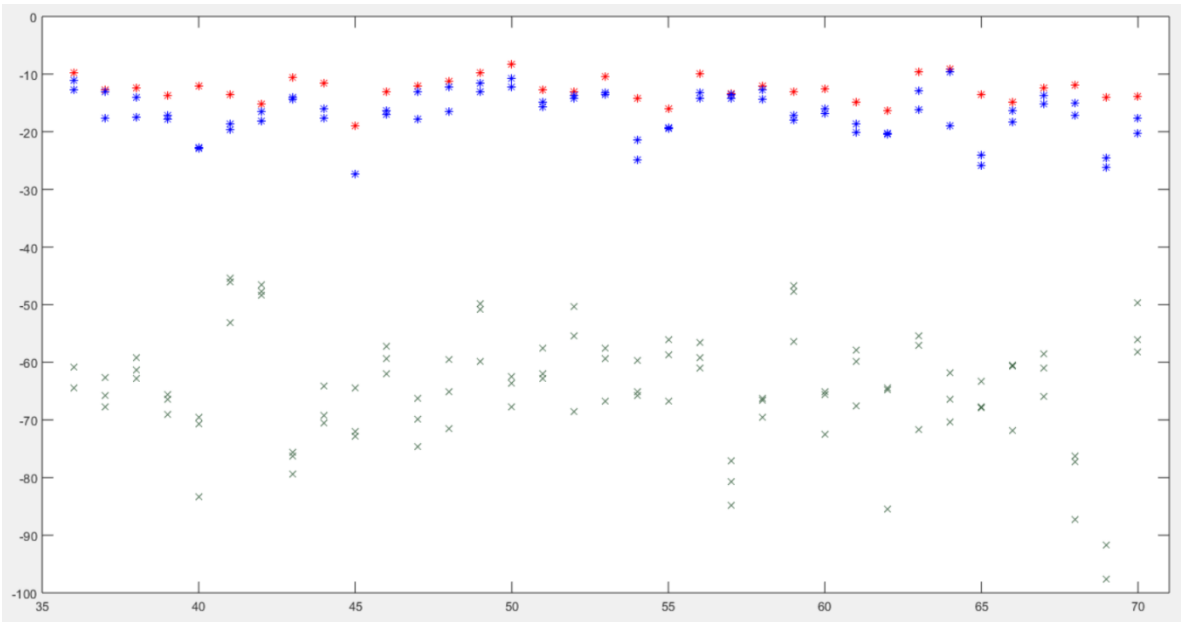

Figura 5.23. Resultados con un 20% de datos para entrenamiento – parte 2.

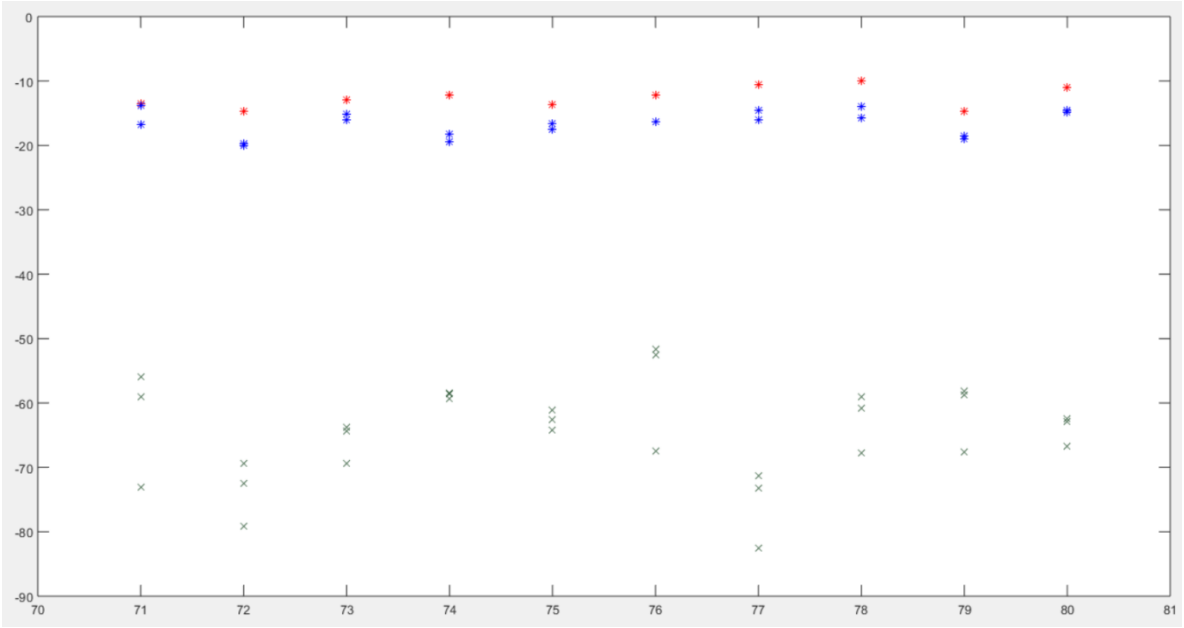

Figura 5.24. Resultados con un 20% de datos para entrenamiento – parte 3.

En el último experimento se presentaron tres errores en la identificación. Al realizar este experimento se puede apreciar cómo hay un aumento en el error de la identificación de los hablantes. El primero se dio entre dos hombres que pertenecen al rango de 18 a 30 años. El segundo fueron dos mujeres, donde la mujer a la que pertenecía la voz era de un rango de edad de 31 a 45 y la que identificó el sistema estaba en al rango de 18 a 30. Por último, la mujer del modelo pertenecía al rango de 46 a 60 años y la identificada por el sistema al de 18 a 30.

Rango de edades de los hablantes y género	
Real	Identificado
Hombre 01-a-01-0001sv.wav 18-30	Hombre 01-a-01-0002sv.wav 18-30
Mujer 02-b-04-0002srv.wav 31-45	Mujer 02-a-04-0005srv.wav 18-30
Mujer 02-c-02-0007srv.wav 46-60	Mujer 02-a-03-0002svr.wav 18-30

Tabla 5.10. Resultados de la identificación, para 20% de datos para entrenamiento.

Después de los cuatro experimentos realizados, se puede observar que el sistema funciona correctamente a pesar de que se encuentran voces de hombres y mujeres en él. Los resultados muestran una cantidad baja en la cantidad de personas identificadas incorrectamente, siendo 3 el número más alto de errores producidos y usando la menor cantidad de información para generar los modelos GMM con sólo el 20%.

A pesar de los errores cometidos, todos fueron entre personas del mismo género y la mayoría entre personas del mismo rango de edad. Esto es un buen indicador de que los algoritmos utilizados

pueden generar resultados bastante positivos, ya que puede discriminar perfectamente entre voces masculinas y femeninas, también de una manera muy buena las diferencias que existen entre las voces de personas del mismo género, pero de diferentes edades.

En la tabla 5.11 se muestran la cantidad de hombres y mujeres que se identificaron en la primera y segunda mitad de la cantidad total, en este caso 80. El resultado ideal es que, entre los identificados, si se trata de un hombre, los primeros 40 valores obtenidos correspondan a puros hombres y los últimos 40 a mujeres. En caso de que se trate de una mujer sería el caso contrario, los primeros 40 correspondientes a mujeres y el resto a hombres.

Esto nos indicaría que el sistema discriminó perfectamente las voces masculinas de las femeninas, ya que los primeros valores corresponden a los valores más altos obtenidos.

20%			
hombres		mujeres	
1 a 40	41 a 80	1 a 40	41 a 80
19	21	21	19
19	21	21	19
20	20	20	20
22	18	18	22
18	22	22	18
18	22	22	18
19	21	21	19
20	20	20	20
⋮	⋮	⋮	⋮
18	22	22	18
17	23	23	17
18	22	22	18
18	22	22	18
20	20	20	20
18	22	22	18
17	23	23	17
18	22	22	18

Tabla 5.11. Cantidad de hombres y mujeres en cada mitad.

5.2 Experimentos disminuyendo únicamente los datos de entrenamiento

En estos experimentos se utilizaron las mismas grabaciones que en el anterior, la diferencia que fue que se mantuvo el 50% de la grabación para pruebas y para el entrenamiento se redujo 10%, esto quiere decir, que se utilizó 40%, 30% y 20%.

Los resultados para el experimento con 40% de datos para el entrenamiento se obtuvo un 100% de positivos en la identificación. La gráfica muestra los datos de los valores del log-likelihood, como se puede observar, existen valores que se mantienen a una distancia considerable y algunos otros se ven muy cercanos.

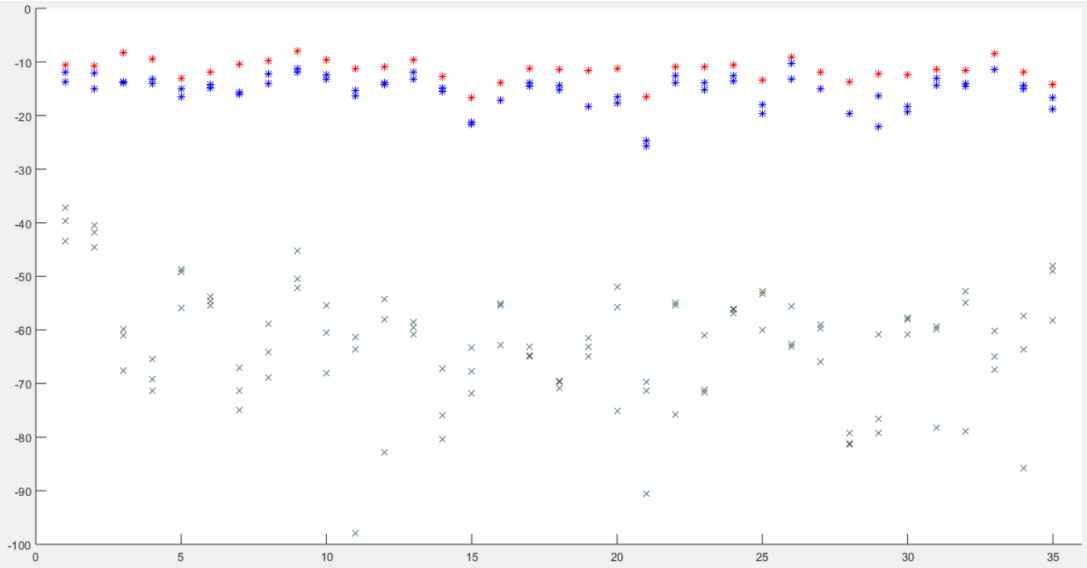

Figura 5.25. Resultados con un 40% de datos para entrenamiento – parte 1.

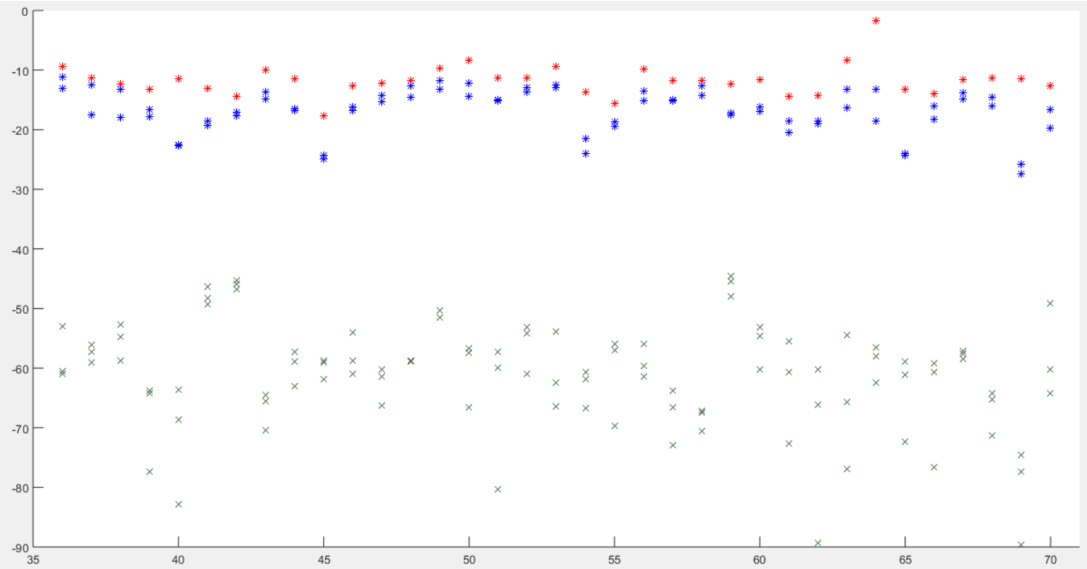

Figura 5.26. Resultados con un 40% de datos para entrenamiento – parte 2.

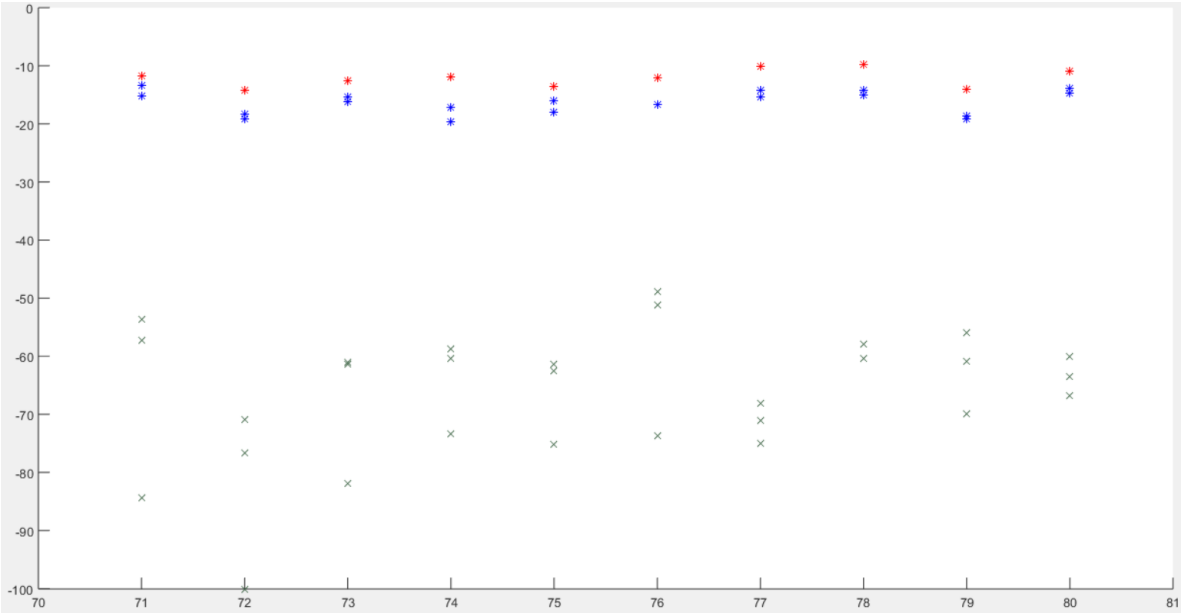

Figura 5.27. Resultados con un 40% de datos para entrenamiento – parte 3.

Los resultados obtenidos haciendo uso del 30% de la grabación para la generación de los modelos, fueron los mismos que con 40%, un 100% de resultados correctos en la identificación.

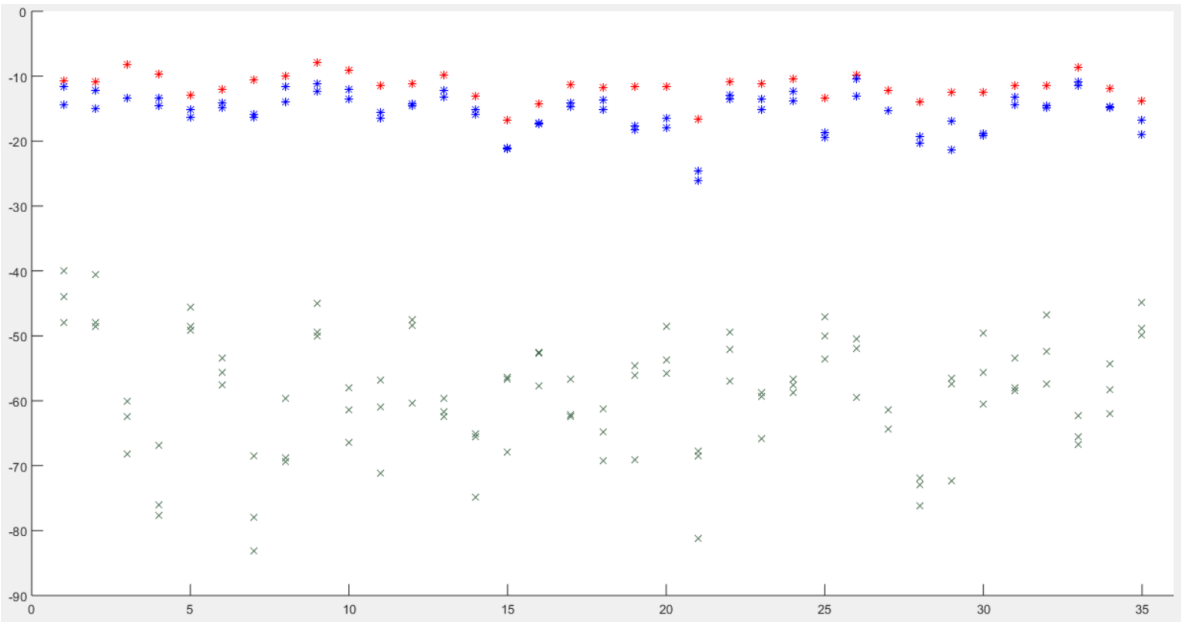

Figura 5.28. Resultados con un 30% de datos para entrenamiento – parte 1.

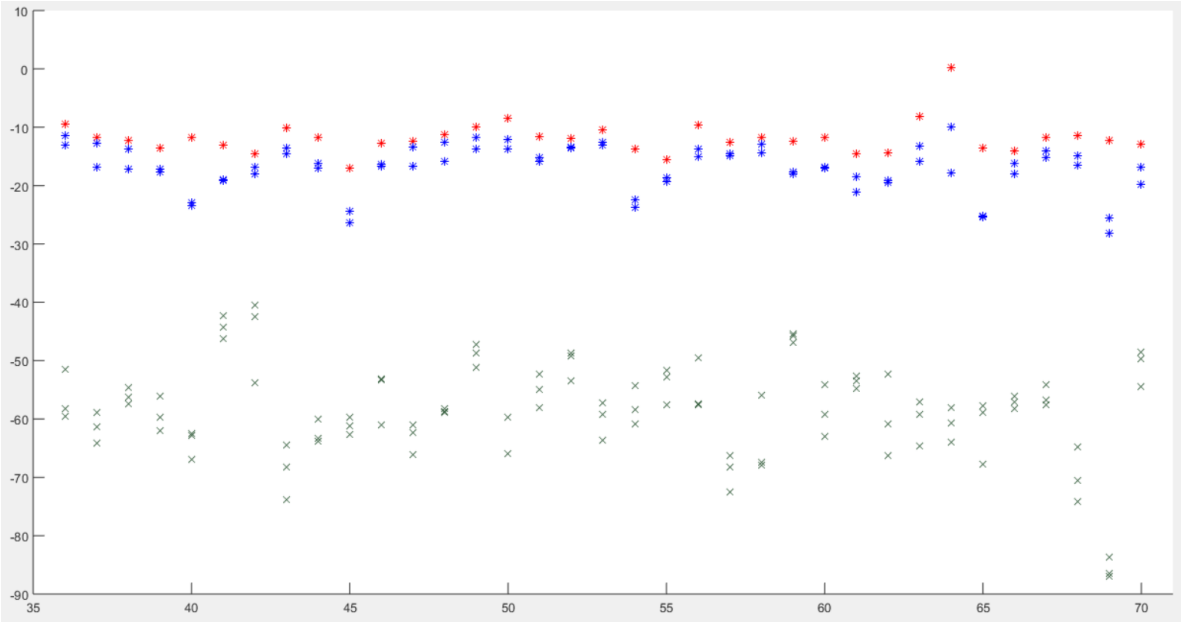

Figura 5.29. Resultados con un 30% de datos para entrenamiento – parte 2.

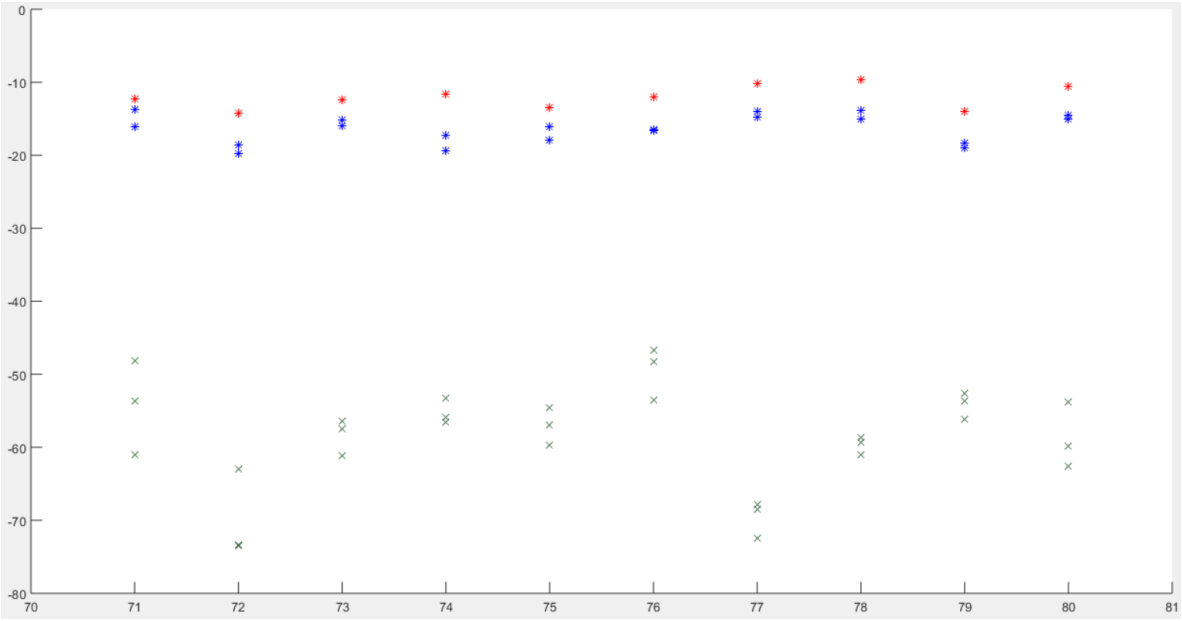

Figura 5.30. Resultados con un 30% de datos para entrenamiento – parte 3.

En las gráficas anteriores se puede observar que las distancias entre el punto más alto, que es el valor correspondiente al hablante reconocido, no son muy grandes con respecto al segundo punto.

El último de los experimentos se utilizó únicamente un 20% de información en la etapa de entrenamiento y el mismo 50% para la de prueba.

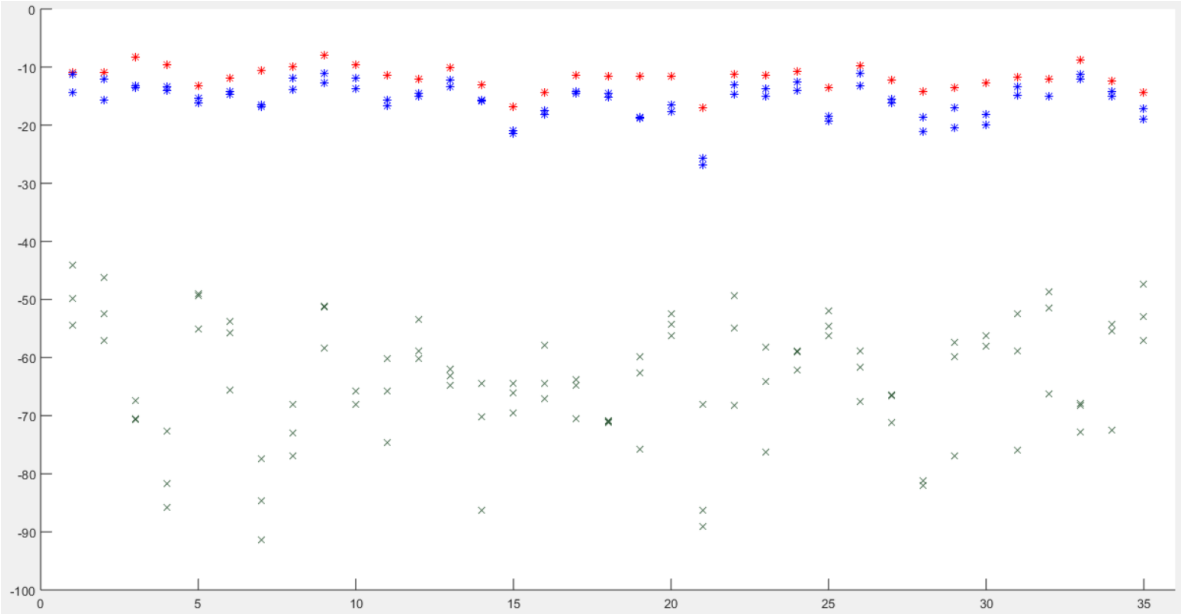

Figura 5.31. Resultados con un 20% de datos para entrenamiento – parte 1.

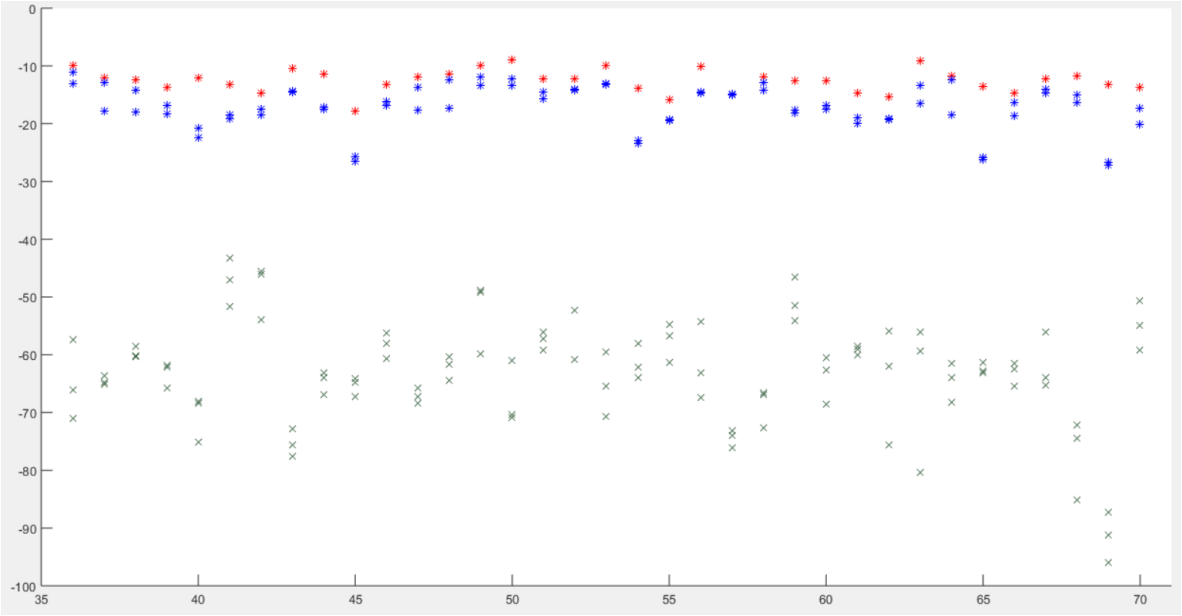

Figura 5.32. Resultados con un 20% de datos para entrenamiento – parte 2.

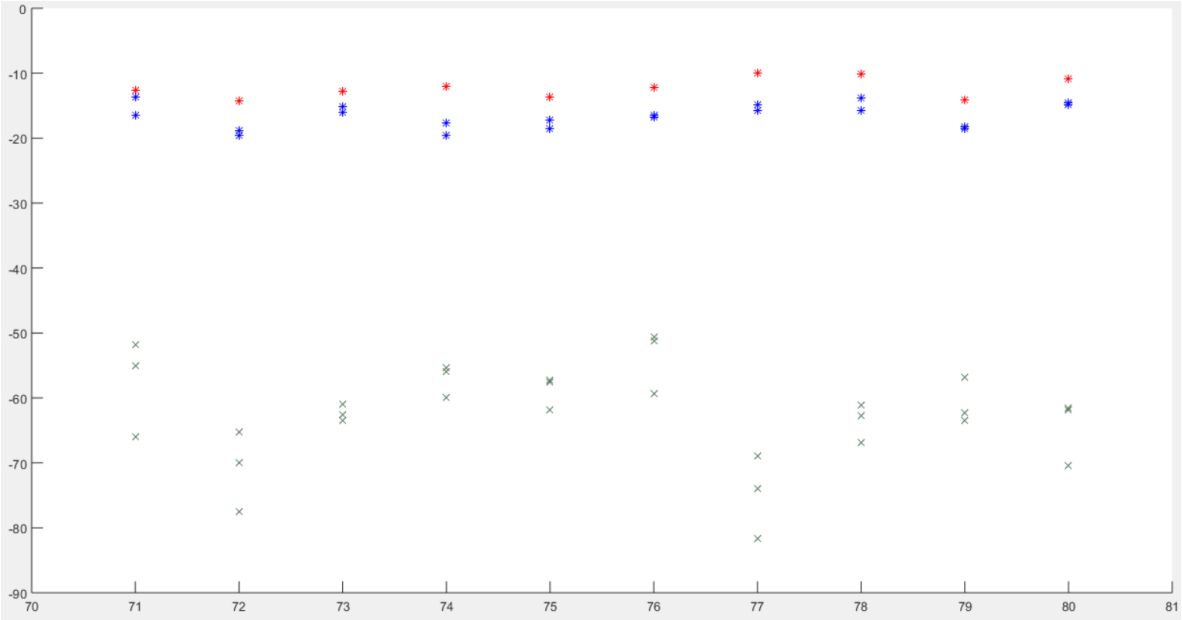

Figura 5.33. Resultados con un 20% de datos para entrenamiento – parte 3.

En este último experimento los resultados fueron un poco menores en cuanto a resultados positivos se obtuvo un 97.5%, esto se traduce en 2 errores de identificación y se muestran en la tabla 5.12. Como se puede observar en la tabla, el primer error sucedió en una mujer del rango de edades de 31 a 45 y el tipo de llamada es celular-fijo; la persona identificada fue otra mujer del rango de edades de 18 a 30 con el mismo tipo de llamada. El segundo error es una mujer de edad entre 46 y 60, el tipo de llamada es público -fijo, en este caso se identificó una mujer de edad 18 a 30 y llamada público-cel.

Rango de edades de los hablantes y género	
Real	Identificado
Mujer 02-b-04-0002srv.wav 31-45	Mujer 02-a-04-0003srv.wav 18-30
Mujer 02-c-02-0007srv.wav 46-60	Mujer 02-a-03-0002svr.wav 18-30

Tabla 5.12. Cantidad de hombres y mujeres en cada mitad.

Capítulo 6. Conclusiones

En el primer experimento donde se realizaron pruebas únicamente con individuos del mismo sexo, los resultados arrojan un porcentaje elevado en las personas identificadas correctamente, tanto para los hombres como para las mujeres.

Como se puede observar en las tablas, los resultados indican que el sistema es robusto, en el sentido que mantiene una precisión muy elevada en condiciones de al menos 30% del corpus en entrenamiento, que equivale a 54 segundos, aproximadamente. En el caso de un entrenamiento de 20% los resultados bajan para las mujeres a 97.5% y se mantienen para los hombres al 100%. Por otro lado, los resultados muestran que los errores ocurrieron entre personas del mismo sexo, en este caso mujeres, esto indica que el sistema no tuvo problema en poder discriminar las voces de los hombres y las mujeres.

Los resultados en general son bastantes positivos, a pesar de que los archivos contienen una gran cantidad de ruido ambiental, en especial aquellos que fueron obtenidos en plena vía pública. Esto demuestra que los algoritmos funcionaron de forma correcta y crearon modelos lo suficientemente capaces de reconocer las características de los hablantes, lo cual permite cumplir con el objetivo principal que era la validación de la efectividad de las técnicas en la tarea de reconocimiento de hablantes.

La ventaja principal que ofrece este toolkit es que se puede ver todo el proceso de manera transparente, incluso los valores arrojados por el algoritmo MLE para la identificación final. Cuenta con comentarios puntuales en cada línea de código que ayuda a comprender el paso en el que te encuentras y ejemplos de cómo utilizar cada una de las funciones.

En la mayoría de los casos, desconocemos el audio con el que se entrenan los sistemas o se utilizan grabaciones hechas en ambientes controlados. El Corpus Valquiria cuenta con grabaciones hechas en la vida real, ya sea dentro de una casa o inlcuso con todo el ruido externo que se puede presentar cuando una persona se encuentra caminando. Los resultados obtenidos siguen siendo de alta tasa de éxito en las identifiaciones, incluso en estas condiciones más complicadas, da visibilidad que este algoritmo puede ser utilizado con "gold standard" para futuras tecnologías por desarrollarse.

En el ámbito forense, se puede desarrollar alguna aplicación móvil en la que este algoritmo sea capaz de presentar identificaciones en situaciones de extorsión. Podría ayudar a distinguir la voz del extorsionador y debido a que es un algoritmo que no implica un procesamiento tan grande como los que se desarrollaron después, es ideal para ejecutarse en dispositivos que no tengan los recursos de cómputo más altos.

Al estar enfocado en un ambiente forense, la utilización de grabaciones cortas para la etapa de entrenamiento era primordial, porque nos permite tener la seguridad de que el resultado obtenido no será malo a pesar de la falta de datos más amplios.

Es claro que las técnicas con el paso del tiempo van mejorando y desarrollándose nuevas. Actualmente hay métodos que empiezan a tomar fuerza como las redes neuronales y el llamado deep learning, gracias a éstas llegaremos al punto en el que los sistemas computacionales puedan alcanzar un nivel de efectividad aún mejor del que se tiene en este momento.

Referencias

[1] Sadoki furui, “50 years of progress in speech and speaker recognition”, 2005

[2] B. S. Atal, “Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification”, 1974

[3] G. R. Doddington, “Speaker recognition—Identifying people by their voices”, 1985

[4] F. K. Soong, ”A vector quantization approach to speaker recognition”, 1985

[5] Younès Bennani, “Text-independent talker idetification system combining connectionist and conventional models”, 1992

[6] Douglas A. Reynolds, “Speaker identification and verification using Guassian mixture speaker models”, Speech Communication Journal, volume 17, pp. 91-108, 1995

[7] J. M. Colombi, “Cohort selection and word grammar effects for speaker recognition”, 1995

[8] Joseph P. Campbell, “ Speaker Recognition: A Tutorial”, 1997

[9] John H.L. Hansen, “Speaker Recognition by Machines and Humans”, Signal Processing Magazine,Volume 32 number 6, November 2015, pp.74-99

[10] D.A. Reynolds, T.F. Quatieri and R.B. Dunn, “Speaker Verification Using Adapted Gaussian Mixture Models”, Digital Signal Processing Journal, Volume 10, Issues 1–3, January 2000, Pages 19-41, ISSN 1051-2004.

[11] W. M. Campbell, D. E. Sturin, D. A. Reynolds, A. Solomonoff, “SVM based speaker verification using a GMM supervector kernel and NAP viability compensation”, 2006

[12] https://www.nist.gov/sites/default/files/documents/itl/iad/mig/SRE16_Eval_Plan_V1-0.pdf

[13] Michel Vacher, Benjamin Lecouteux, Javier Serrano-Romero, Moez Ajili, François Portet, Speech and Speaker Recognition for Home Automation: Preliminary Results. 8th International Conference Speech Technology and Human-Computer Dialogue ”SpeD 2015”, Oct 2015, Bucarest, Romania. pp.181-190.

[14] Brunet, Kevin & Taam, Karim & Cherrier, Estelle & Faye, Ndiaga & Rosenberger, Christophe. (2013). Speaker Recognition for Mobile User Authentication: An Android Solution. 8ème Conférence sur la Sécurité des Architectures Réseaux et Systèmes d'Information (SAR SSI).

[15] Hossein Salehghaffari, "Speaker Verification using Convolutional Neural Networks", Control/Robotics Research Laboratory (CRRL), Department of Electrical and Computer Engineering, NYU Tandon School of Engineering (Polytechnic Institute), NY 11201, USA

[16] Javier Ortega-Garcia, Joaquin Gonzalez-Rodriguez, Victoria Marrero-Aguiar, “AHUMADA: A large speech corpus in Spanish for speaker characterization and identification”, Speech Communication 31, 2000, pp. 255-264

[17] V. Tiwari , “MFCC and its applications in speaker recognition”, International Journal on Emerging Technologies, volume 1, pp. 19-22, 2010.

[18] Lindasalwa Muda, Mumtaj Begam and I. Elamvazuthi”, Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques”, Journal Of Computing, Volume 2, Issue 3, March, 2010, pp. 138-143, ISSN 2151-9617

[19] Leena R Mehta , S.P.Mahajan , Amol S Dabhade, “Comparative Study Of MFCC And LPC For Marathi Isolated Word Recognition System”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 2, Issue 6, June 2013, pp. 2133-2139

[20] J. Myung, “Tutorial on maximum likelihood estimation”, Journal of Mathematical Psychology , volume 47, pp. 99-100, 2003.

[21] J. Orloff and J. Bloom, “Maximum Likelihood Estimates”, Class 10,18.05, Spring course 2014, http://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/readings/MIT18_05S14_Reading10b.pdf

[22] J.C. Watkins, “Maximum likelihood Estimation”, Introduction to Statistical Methodology, November 2011, http://math.arizona.edu/~jwatkins/o-mle.pdf

[23] D.A. Reynolds and R.C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," in IEEE Transactions on Speech and Audio Processing, vol. 3, no. 1, pp. 72-83, Jan 1995.

[24] Tomi Kinnunen, Haizhou Li, “An overview of text-independent speaker recognition: From features to supervectors”, Speech Communication 52, pp. 12-40, 2010

[25] A. B. Poritz, “Linear predictive hidden Markov models and speech signal”, ICASSP 1982, pp. 1291-1294, May 1982

[26] Alvin F. Martin, Mark A. Przybocki. “NIST 2003 Language Recognition Evaluation”, EUROSPEECH 2003 – GENEVA. Pp 1341-1344, September 2003