Universidad
Carlos III de Madrid

Trabajo Fin de Grado

Diseño e Implementación de un Sistema de Reconocimiento de Hablantes

Raquel Donoso García del Castillo
Grado en Ingeniería de Sistemas Audiovisuales

Tutor: Julio Villena Román

Leganés, febrero de 2014

Título: Diseño e Implementación de un Sistema de Reconocimiento de Hablantes

Autora: Raquel Donoso García del Castillo

Tutor: Julio Villena Román

EL TRIBUNAL

Presidenta: Iria Manuela Estévez Ayres

Vocal: Almudena Lindoso Muñoz

Secretario: Víctor Elvira Arregui

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día 04 de marzo de 2014 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO PRESIDENTE

Agradecimientos

En primer lugar agradezco la atención de mi tutor Julio Villena, que puso empeño en proporcionarme un tema que fuera de mi interés el cual pudiera resultar atractivo además de ampliar mis conocimientos.

Debo agradecer enormemente también el apoyo de mi familia, amigos y en especial de David Villatobas, por estar ahí en todo momento y confiar en mí incluso en los momentos más duros.

Finalmente debo hacer una mención especial para mis compañeros que han aguantado mi carácter y han hecho de los estudios algo más llevadero y gratificante. Muy en detalle dar las gracias a Aroa Bellés y Marta Chacón por no desesperarse nunca conmigo, saber trabajar en equipo y crear ese ambiente agradable que me hizo seguir hacia delante con la carrera. Siempre es reconfortante saber que, además de una buena formación, te quedan amigos de por vida.

RESUMEN

Un sistema de reconocimiento de hablantes se resume como aquel capaz de reconocer automáticamente qué persona está hablando de entre las personas pertenecientes a una base de datos de señales de audio previamente clasificadas. Para construir estos sistemas se utiliza la información conocida sobre el tracto vocal de un individuo y las propiedades físicas del sonido.

Estos sistemas se pueden utilizar para innumerables aplicaciones y en numerosos ámbitos. A continuación algunos ejemplos:

- **Seguridad:** Autenticación o prevención de fraude.
- **Comercio:** Confirmación de compras a distancia.
- **Educación:** Nuevas enseñanzas o métodos de aprendizaje para personas discapacitadas.

En este trabajo se planteará un sistema de reconocimiento de hablante implementado sobre un código de software matemático programado en Matlab.

El sistema permite reconocer a un locutor entre una base de datos existente con la cual se ha entrenado al clasificador. Las características principales que se extraen de los audios son los coeficientes Cepstrum, que aportan numerosa información sobre pequeños fragmentos de audio.

Además, se incluye en detalle la explicación sobre la obtención de estos coeficientes y sobre la elección de todas las variables o procedimientos utilizados en el algoritmo.

Finalmente, se incluyen los experimentos realizados, junto a las conclusiones y el presupuesto final del proyecto.

Palabras clave: Reconocimiento, hablante, locutor, sistema, Cepstrum.

ABSTRACT

A speaker recognition system is a process that recognizes the person who is talking among those stored in a database of audio signals previously classified. The system uses information about the vocal tract and the physical properties of sound.

These systems can be used for a lot of applications and a lot of fields. Some examples:

- **Security:** Authentication or fraud prevention.
- **Trade:** Confirmation about distance shopping.
- **Education:** New methods of teaching or disabled people.

This document presents a speaker recognition system implemented on a mathematical programming code in Matlab.

The system can recognize a speaker from an existing database that has been used to training. The main features that are extracted from the audios are the Cepstrum coefficients, which provide a lot of information about small pieces of audio.

Also it includes explanation of the coefficients and variables and procedures used in the algorithm.

Finally, experiments, conclusions and final project budget are included.

Keywords: Recognition, speaker, system, Cepstrum.

ÍNDICE GENERAL

ÍNDICE DE ECUACIONES

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Capítulo1: INTRODUCCIÓN

1.1. Introducción

La inteligencia artificial fue introducida en el estudio científico en el año 1950 por el inglés Alan Turing, provocando un interés mundial con su pregunta "*Can machines think?".* Numerosos científicos comenzaron sus investigaciones orientadas a crear modelos y teorías que dieran explicación al funcionamiento de la inteligencia. En la actualidad los estudios de inteligencia artificial se basan en el desarrollo de sistemas de procesado de datos que imitan el comportamiento de la mente humana, con sistemas de decisión tras un aprendizaje previo [INT].

Si bien un gran número de investigadores buscan respuestas en el estudio psicológico, la gran mayoría se apoya en la física para conseguir teorías aproximadas y asimilar sus sistemas al comportamiento humano (el ojo, el oído, ondas, frecuencias, etc.).

La voz humana es definida por la Real Academia de la Lengua como *"Sonido que el aire expelido de los pulmones produce al salir de la laringe, haciendo que vibren las cuerdas vocales"* o *"Calidad, timbre o intensidad de este sonido"* [RAE]. El habla consiste en combinar las unidades fónicas (fonemas) que la voz humana es capaz de generar para formar una lengua. Estos símbolos no son idénticos de un individuo a otro, pero poseen características comunes que los hace descifrables dentro de una lengua o dialecto.

Palabra	Fonemas
patata	/p/ /a/ /t/ /a/ /t/ /a/
hola	/o/ /l/ /a/
escribir	/e/ /s/ /k/ /r/ /i/ /b/ /i/ /r/

Tabla 1.1.1: Relación palabra – fonemas.

Las cuerdas vocales son los pliegues de nuestro aparato fonador que generan la voz. Estos pliegues son dos cuerdas enfrentadas con forma de válvula, que abren o cierran el paso del aire a la altura de la garganta. Cuando las cuerdas vocales se encuentran en estado de reposo dejan entrar y salir el aire libremente, mientras que cuando se contraen obstaculizan el paso del aire, lo que las hace vibrar generando el sonido.

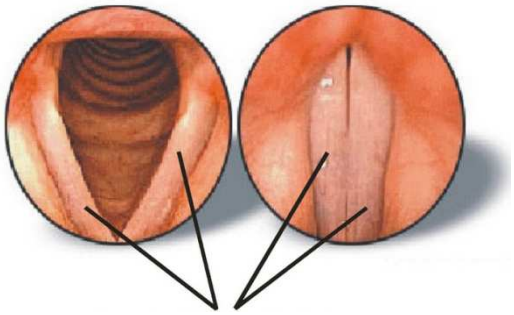

Figura 1.1.2: Cuerdas vocales abiertas y cerradas.

El reconocimiento de voz ha sido uno de los campos más estudiados en la inteligencia artificial, buscando como finalidad la comunicación entre personas y máquinas mediante lenguaje hablado. La ciencia ha avanzado enormemente con este propósito implantando estos sistemas en gran cantidad de dispositivos, como por ejemplo teléfonos móviles o computadoras, obteniendo buenos resultados.

Ahora bien, si dichas máquinas están capacitadas para extraer el mensaje y la información del texto y procesarlos según las necesidades, aún no son capaces de distinguir si el orador es una persona u otra con exactitud. De ahí nace la necesidad de crear nuevos algoritmos, no tan enfocados a interpretar mensajes sino a distinguir de quién provienen los mismos. Estos sistemas son conocidos como Speaker recognition systems o Sistemas de reconocimiento de hablantes.

Actualmente los sistemas pueden dividirse en dos grandes ramas de estudio, la autentificación biométrica, que trata de verificar la identidad del hablante, y el estudio bajo entrenamiento, que está orientado a reconocer la voz de un individuo entre un catálogo de voces ya estudiadas y analizadas previamente.

Estos algoritmos en conjunto tienen numerosas aplicaciones. La más difundida es el control de acceso dictado por voz, útil para la confirmación de compras telefónicas, transacciones bancarias, acceso a bases de datos o acceso remoto a maquinaria por ejemplo. Comercialmente se utiliza en aplicaciones de domótica, para el comando de sillas de ruedas o el control de robots y también son usados en el campo de seguridad

para el rotulado de interlocutores en una conversación grabada o en investigaciones policiales que involucran muestras grabadas de voz.

1.2 Motivación

El ser humano necesita comunicarse. Desde el principio de los tiempos la raza humana ha diseñado y optimizado lenguajes de signos con los que poder comunicarse con el medio en el que vive. Estos lenguajes son diferentes dependiendo de sus costumbres, su geografía y en general su forma de vida.

A día de hoy, las máquinas se encuentran muy integradas en la sociedad, utilizándose para un sinfín de aplicaciones de numerosos campos que nos permiten una mayor calidad de vida en cuanto a medicina, esfuerzo y comodidad. Es por esto que el ser humano necesita seguir evolucionando su lenguaje hasta amoldarlo a sus necesidades, encontrando muy atractivo el lenguaje oral por su uso natural y su simplicidad. A nivel computacional, conseguir que las máquinas entiendan este lenguaje es una ardua tarea sobre la que aún queda mucho por trabajar y experimentar.

Influye también el factor social que se aparece dividido fuertemente en dos ideologías. Por una parte se encuentra la ambición y la búsqueda de comodidad que infligen al ser humano una necesidad de evolucionar y crear nuevas tecnologías, mientras que opuestamente existe un miedo generalizado por perder los valores humanos y aparece la sospecha de que las máquinas cambian nuestra forma de vida a peor.

Sin embargo, es un hecho incuestionable que la tecnología ha llegado para quedarse y hay que aceptar que ya forma parte de nuestro hábitat y nuestra sociedad. Las posibilidades que ofrece son infinitas y no tiene sentido oponerse al progreso igual que no habría tenido sentido oponerse al conocimiento de la física.

La motivación principal de este trabajo, en mi caso, es poder crear sistemas que mejoren la vida de personas con discapacidades, como pueda ser una persona con problemas auditivos o malformaciones en las manos, para los cuales sería de gran ayuda hacer uso de su voz y así poder manejar dispositivos o máquinas como el resto de la gente sin que sus limitaciones supongan un problema.

1.3 Objetivo

Este trabajo engloba un estudio sobre un sistema de reconocimiento de hablantes basado en entrenamiento previo, utilizando las características que ofrecen los coeficientes Cepstrum y usando el algoritmo KNN (K Nearest Neighbors) para su clasificación. El objetivo principal será en todo momento la clasificación de diferentes audios atendiendo a su similitud con una matriz compuesta por coeficientes cepstrum de un conjunto de personas diferentes.

El objetivo del Trabajo Fin de Carrera es, por tanto, adquirir los conocimientos necesarios para diseñar un sistema de estas características e implementarlo sobre un lenguaje de programación. Para ello será indispensable la investigación de algoritmos y técnicas que se ven implícitos en el diseño así como su correcta evaluación mediante un conjunto de experimentos que determinará si el sistema es apto para la causa.

1.4 Estructura del trabajo

La estructura del trabajo comenzará por un planteamiento específico del problema a tratar, con sus exigencias y restricciones, seguido por la descripción en detalle del diseño técnico utilizado y finalizando con los resultados, presupuesto, conclusiones, y trabajos futuros pendientes.

Capítulo 2: PLANTEAMIENTO DEL PROBLEMA

A continuación se detalla el contexto donde se encuentra enmarcado este proyecto, dando en primer lugar un ligero repaso al marco histórico para posteriormente extender la definición de un sistema de reconocimiento de hablante.

Más tarde se detallarán las normativas existentes, los requisitos y las restricciones a la hora de llevarlo a cabo.

2.1 Análisis del estado del Arte

Como en cualquier estudio, lo primero para empezar a trabajar es tener un conocimiento previo sobre el tema a tratar, tanto de su historia como de los trabajos e investigaciones actuales. Si bien el trabajo a realizar debe servirnos para sacar nuestras propias conclusiones y obtener nuevos resultados, conocer datos anteriores puede ser de gran utilidad a la hora de no cometer los mismos errores y tener una vaga idea de por dónde enfocar el trabajo.

En la segunda mitad del siglo XX (alrededor de los años 50) se realizaron los primeros intentos de construir una máquina que realizara tareas de reconocimiento mediantes los principios fundamentales de la fonética acústica. El primer sistema documentado se remonta a 1952, a manos de K. Davis, R. Biggulph y S. Balashek. Su dispositivo permitía identificar a un solo hablante que realizaba pronunciaciones de los diez dígitos de forma aislada [DAV]. La máquina se estaba basada en medidas de resonancias espectrales de tracto vocal para cada dígito, obteniéndose medidas mediantes el uso de bancos de filtros analógicos.

En 1959, en el University College de Londres, P. Denes desarrolló un sistema que reconocía de forma aceptable cuatro vocales y nueve consonantes, mediante el uso de información estadística acerca de las secuencias válidas de fonemas en inglés, no utilizado hasta entonces. Nace el objetivo de reconocer unidades léxicas con dos o más fonemas, incorporando conocimiento lingüístico en los sistemas [RAB].

En los años 60 se generaliza el uso de ordenadores para el reconocimiento del habla y aparecen tres grandes avances que pueden considerarse los pilares de la experimentación actual:

- T. Martin, A. Nelson y H. Zadell se centran en desarrollar soluciones realistas para los problemas que supone la falta de uniformidad de las escalas de tiempo en los hechos de habla. Se diseñaron métodos elementales de normalización de tiempo [RAB].
- T.K Vintsyuk estudia la utilización de métodos de programación dinámica para conseguir el alineamiento temporal de los pares de realizaciones de habla [JUA].
- D. R. Reddy trabajó en el habla continua mediante el seguimiento dinámico de fonemas, reconociendo oraciones dependientes del hablante sobre 561 palabras [JUA].

En 1971 se comienza a tratar el reconocimiento de hablante paralelamente a las investigaciones de reconocimiento de voz. F. Itakura aplica técnicas LPC (Linear Predictive Coding) mediante el uso de medidas de distancia adecuadas sobre un conjunto de parámetros espectrales. Los científicos buscaban las características en parámetros estadísticos o predictivos, matrices de covarianza, histogramas de frecuencia, etc. [FUR].

S. Furui propone utilizar la combinación de coeficientes espectrales como características primordiales para el reconocimiento de hablante pero nadie trató de utilizarla hasta muchos años después a finales de los 80. Este método se utiliza ahora a efectos prácticos en casi todos los sistemas de reconocimiento de voz.

En 1996 Matsui pone en funcionamiento un sistema en que las frases a testear cambian por completo cada vez que éste se utiliza. El sistema aceptaba la expresión de entrada sólo cuando determinaba que el locutor había pronunciado dicha frase. El método rechazaba enunciados con texto diferente al estudiado además de reconocer al locutor con un vocabulario limitado [MAT].

En los últimos años se diferenciaron los estudios de voz entre los que perseguían la identificación del locutor, y los que se centraban en verificar al mismo. Centrándonos en la verificación de hablante, hay dos tipos de estudios atendiendo a sus modos de entrada [REC]:

- **Texto dependiente**.
  El hablante provee muestras de voz que se corresponden con el mismo texto en las fases de entrenamiento y reconocimiento. Las contraseñas son fijas y basta con alinear las muestras con los patrones temporalmente y estudiar la similitud.

- **Texto independiente**.
  El hablante aporta diferentes textos que no tienen por qué coincidir con los usados para el entrenamiento.

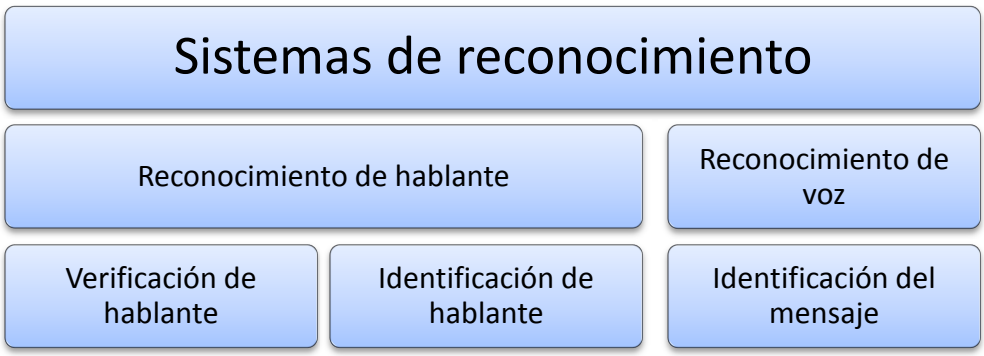

Figura 2.2.1: Esquema sistemas de reconocimiento.

El reconocimiento de hablantes es un sistema que persigue reconocer automáticamente la persona que está hablando teniendo como base un conjunto de voces entrenadas previamente que se utiliza como patrón. Es un principio diferente a la verificación de hablantes, que acepta o rechaza la sentencia de si el hablante es quién dice ser. Los sistemas de reconocimiento de hablante se componen en general de tres fases descritas a continuación:

En primer lugar los audios examinados necesitan siempre un pre-procesado. Para hacer un buen trabajo se debe tener en cuenta que la duración de los mismos sea similar, así como procurar que las grabaciones tengan una calidad aceptable. Si el audio está contaminado con ruidos, contiene música de fondo, está grabado en un ambiente donde hay más personas hablando a un nivel inferior o simplemente contiene sonidos que no pertenecen al orador que se va a examinar, es probable que aparezcan errores y se dificulte la elección. Por ello el pre-procesado es importante antes de realizar cualquier otro paso. Se utilizan filtros para eliminar ciertos ruidos o frecuencias, así como algoritmos basados en la energía de la señal para eliminar períodos donde la persona está en silencio y que no interesan analizar a posteriori. Dependiendo de las señales de entrada, puede ser necesario también aplicar un cambio de frecuencia de muestreo o cualquier otra operación similar que dé como salida un conjunto de audios con similares características técnicas.

La segunda técnica importante es la extracción de características. El objetivo en cualquier experimento que requiera una extracción de características es seleccionar un conjunto de variables a partir de un conjunto inicial que optimice distintas subclases con información. El último matiz es interesante, ya que no todas las características aportan información útil a un experimento. Se debe hacer un estudio previo al tema analizado y hacer diferentes pruebas hasta conseguir un conjunto de características relevantes, que podrá ser mayor o menor número en función de la complejidad del problema.

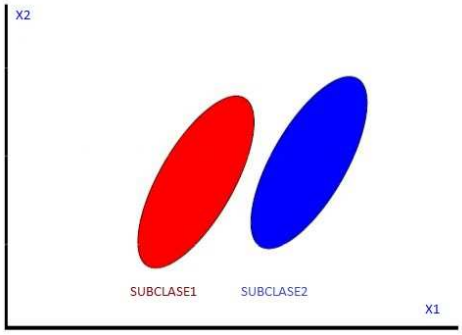

Figura 2.6.2 Subconjuntos tras extracción de características.

De la misma manera, puede ocurrir que la elección de las variables originales sobre sus ejes no den un buen resultado, pero este mejore de forma notable creando una nueva variable que dependa de las anteriores.

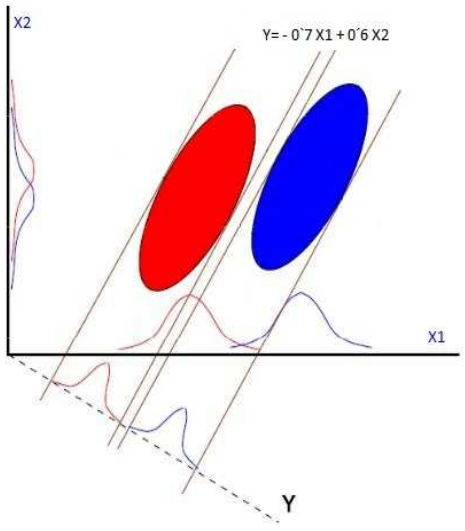

Figura 2.4.6: Extracción de características con combinación lineal.

Una de las técnicas más utilizadas en la extracción de características es la PCA (Principal Component Analysis), que consiste en transformar el espacio de representación de subclases en uno nuevo donde los datos se encuentren incorrelados, de forma que alguna dirección, la varianza de los datos sea máxima. Para definir los nuevos ejes se definen como aquellos que son perpendiculares a los anteriores siendo su dirección la máxima varianza entre todos los ejes posibles.

La tercera y última fase es la clasificación de vectores. Una clasificación consiste en realizar un algoritmo que asigne un elemento no catalogado en una categoría conocida. Para ello el clasificador dispone de una o varias reglas definidas que se obtienen de un conjunto de datos bien etiquetados.

Centrándonos en el reconocimiento de hablante, los clasificadores más usados son los GMM (Gaussian Mixture Models), ANN (Artificial Neural Networks), HMM (Hidden Markov Models) y SVM (Support Vector Machines) [PAD].

La técnica más empleada actualmente es la HMM, que ha supuesto avances importantes, aunque aún no se ha conseguido implementar un sistema de reconocimiento de alto rendimiento.

La otra técnica con la que se obtienen buenos resultados, y que será utilizada en este proyecto, son los métodos SVM. Estos algoritmos maximizan las distancias entre las muestras permitiendo mayor robustez ante el ruido y evitando los problemas de convergencia que suponen las redes neuronales artificiales. Para que estos algoritmos funcionen se necesita una buena extracción de características principales, dando un rendimiento óptimo si se combina con un análisis cepstral, el cual estudiaremos más adelante [PAD].

2.2 Estado del mercado

Actualmente el mercado de sistemas de reconocimiento es muy amplio y variado. Las empresas se han interesado más por el reconocimiento de voz ya que estos sistemas tienen un gran número de aplicaciones más atractivas para el usuario de a pie. Muchos dispositivos de uso corriente como por ejemplo los teléfonos móviles ya incorporan sistemas para introducir texto mediante voz, y muchas instituciones

como pequeñas y grandes empresas, universidades o salas de conferencias utilizan el reconocimiento de voz para dejar actas por escrito de forma automática.

Algunas de las empresas que proveen estos sistemas se detallan a continuación.

- NUANCE Communications

Nuance ofrece el software de reconocimiento de voz *Dragon Enterprise* que permiten manejar cualquier equipo por voz, crear instrucciones de voz para automatizar procesos empresariales o la transcripción mediante dictado oral de cualquier texto. En los últimos años, Nuance ha aplicado su experiencia en el reconocimiento del habla dentro del mercado emergente de las interfaces de voz. La empresa provee tecnología de reconocimiento de voz para muchas otras empresas.

En 2016 ha lanzado al mercado también el software *Nina*, implementado en los dispositivos Android. *Nina* ofrece un servicio personalizado al cliente para navegar por el teléfono utilizando una interfaz de conversación similar a la humana [NUA].

- APPLE

Siri es el sistema de reconocimiento de voz Apple, uno de los más comercializados y conocidos. No sólo reconoce el mensaje sino que tiene funciones programadas que permiten al programa captar el sentido global y realizar búsquedas en internet referentes al tema hablado. Si no es capaz de entender de qué tema se trata, realiza preguntas al locutor para orientar su búsqueda [APP].

- PHILIPS

Philips ha trabajado en la extensión de flujo de trabajo de reconocimiento de voz *SpeechExec*, un software de reconocimiento de voz que se integra completamente en el flujo de trabajo de una oficina y ofrece funciones esenciales requeridas, como colas automáticas de dictados de varios autores para asistentes predefinidos, utilización del reconocimiento de voz en formularios y plantillas o enrutamiento selectivo de los archivos de dictado a software de reconocimiento de voz o a los asistentes [PHI].

- IBM

ViaVoice es un programa de reconocimiento de voz comercializado por IBM que permite que la computadora transforme en texto las palabras dictadas por una persona. Es necesario un micrófono para realizar la transformación [IBM].

- MICROSOFT

Sapi es una API de libre distribución desarrollada por Microsoft que permite el uso de aplicaciones de reconocimiento de voz dentro de otras aplicaciones de Windows. Todas las versiones del API se han diseñado mediante interfaces accesibles desde muchos lenguajes de programación [MIC].

- GOOGLE

Ok Google es una extensión de búsqueda para dispositivos móviles. Con sólo entrar al navegador y utilizar las palabras “Ok Google” se activa automáticamente permitiendo al usuario realizar cualquier búsqueda a través de voz sin utilizar el teclado [GOO].

- SAIL LABS

Media Mining Indexer procesa el habla de múltiples fuentes en diversos formatos reproduciendo la salida en tiempo real. Es capaz de categorizar los audios en temas específicos y monitorea las noticias de forma simultánea en varios idiomas como inglés, alemán, francés, español, ruso y árabe [SAI].

- PERVOICE

Transcribe es el nombre de la última familia de productos relacionados con reconocimiento de voz que Pervoice ha sacado al mercado. Está especialmente implementado para el idioma italiano pero es competitivo con tecnologías similares para otros idiomas. Además de los requisitos básicos se han desarrollado soluciones que incluyen el estudio de la personalización, la puntuación o el reconocimiento de voz emocional [PER].

2.3 Marco regulador

En la normativa española hay varias leyes que hacen referencia al reconocimiento de voz y que es conveniente conocer a la hora de crear o comercializar un producto.

La Ley de Enjuiciamiento Criminal Español, bajo la rúbrica "De la identidad del delincuente y de sus circunstancias personales" (capítulo III, título V, libro II) comprende una variada gama de actuaciones dirigidas a la identificación del inculpado desde una doble perspectiva: «formal» y «material», en cuanto se trata de determinar tanto «su personalidad, como su posible participación en los hechos objeto del proceso». Esta ley adjunta que:

«Cualquier medio lícito puede ser utilizado a los fines de la investigación, no simplemente los clásicos como testigos, peritos y documentos, sino también los aportados por los modernos avances tecnológicos y científicos, desconocidos en el tiempo de promulgación de la Ley procesal penal, como la grabación de imágenes y

sonidos a través del cinematógrafo, la fotografía, el vídeo, CD, DVD y otros, sin olvidar las identificaciones a través de la dactiloscopia (6), los análisis de voz y los marcadores de ADN extraídos de restos sanguíneos, de semen, de cabellos, de piel, etc. Queda así legalizada, en otras muchas actuaciones investigadoras, la del reconocimiento de voz cuando a través de ella sea posible identificar la del responsable del delito».

En octubre de 2011, treinta y cuatro estados de EEUU prohibieron enviar mensajes de texto mientras se conducía un vehículo y nueve han prohibido ya el uso del teléfono móvil al volante. Sin embargo, estados como Indiana e Illinois han redactado nuevas leyes que prohíben únicamente el uso manual de los dispositivos móviles mientras se conduce. Por tanto, con la función de reconocimiento de voz en estos aparatos, los usuarios pueden crear y responder a correos electrónicos o mensajes de texto mientras conducen sin violar ninguna ley. En España, la Ley de tráfico 19/2001 prohíbe utilizar durante la conducción dispositivos de telefonía móvil excepto cuando la comunicación tenga lugar sin emplear las manos, ni usar cascos auriculares o instrumentos similares [BOE].

Al ser éste un trabajo enfocado en el ámbito de investigación no existen restricciones ni normativas como tal que le atañan.

Habría que valorar la normativa de la LOPD (Ley Orgánica de Protección de Datos) que protege la información privada de las personas, como son un fichero con los datos de los usuarios y grabaciones de sus audios. Otras aplicaciones más genéricas como son reconocer a personajes públicos, y que se han utilizado para llegar a los resultados de este proyecto, no requieren este permiso expreso por ley porque son figuras públicas cuyos datos y audios son públicos.

2.4 Discusión de los datos y propuesta de solución

En este capítulo se ha presentado el dominio de aplicación del proyecto así como diferentes tecnologías que han sido utilizadas y trabajos relativos actuales que resultan interesantes. Si bien estos sistemas no se encuentran aún bien integrados ni aceptados en la sociedad, es destacable que la mayoría de productos comercializados se especializan en los sistemas de reconocimiento de voz. Es por eso que este proyecto irá más enfocado a la rama hermana, el reconocimiento de identificación de locutor por entrada de texto independiente, que tiene también numerosas aplicaciones útiles como veremos más adelante.

Capítulo3: DISEÑO TÉCNICO DE LA SOLUCIÓN

Hay numerosos algoritmos con los que se podría construir un sistema de reconocimiento de hablante. Las variables aparecen desde la elección del lenguaje de programación hasta el tipo de audio, el procesado del mismo, las características estudiadas, el algoritmo de comparación o la distancia utilizada.

Se detallará a continuación el diseño de la solución así como la elección de todos los parámetros citados anteriormente.

3.1 Teoría fundamentada en la voz

El ser humano es capaz de generar voluntariamente ondas de presión acústica a partir de movimientos de la estructura anatómica del sistema fonador humano. La generación de voz comienza en el cerebro con la conceptualización de la idea que se quiere transmitir, la cual se asocia a una estructura lingüística seleccionando las palabras adecuadas y ordenándolas de acuerdo con unas reglas gramaticales. A continuación el cerebro produce impulsos nerviosos que mueven los órganos vocales para producir los sonidos. Los órganos involucrados son las cuerdas vocales, el paladar, la lengua, los dientes, los labios y la mandíbula [BEL].

La frecuencia de cada sonido depende de varios factores, pero principalmente del tamaño y la masa de las cuerdas vocales y de la tensión que se les aplique, lo que hará que el aire salga de los pulmones a una velocidad u otra:

- Mayor tamaño: Menor frecuencia (graves)
- Mayor tensión: Mayor frecuencia (agudos)

La zona que incluye la cavidad faríngea, oral y nasal junto a los elementos articulatorios se denomina cavidad supraglótica mientras que los espacios por debajo de la laringe como la tráquea, los bronquios y los pulmones conforman las cavidades infraglóticas.

El amplio margen de sonidos es posible gracias a que algunos de los elementos de la cavidad supraglótica se controlan a voluntad. La faringe y las cavidades nasal, oral y labial realizan un filtrado modificando el espectro, actuando como resonadores acústicos que enfatizan determinadas bandas de frecuencia reforzando la amplitud de grupos de armónicos situados alrededor de una determinada frecuencia.

Todos los sonidos tienen por tanto variables dependientes del hablante, como el tamaño de las cuerdas vocales, pero también variables en común con otros hablantes, como el rango de frecuencia de un determinado fonema. El conjunto de estas variables hará posible cualquier estudio sobre voz humana.

3.2 Solución empleada

Para implementar el sistema de reconocimiento de hablantes se podría haber elegido entre una amplia variedad de algoritmos. En este caso se ha elegido una combinación que no tiene por qué ser óptima, ya que para hacer esta afirmación se necesita un estudio riguroso de todas las demás, pero que tiene cierto sentido lógico a priori y da unos resultados acordes al objetivo buscado. Se ha seguido la estructura típica de los trabajos de sistemas clasificatorios con reconocimiento de patrones, apareciendo los cuatro bloques principales: procesado, extracción de características, algoritmo de clasificación y toma de decisión.

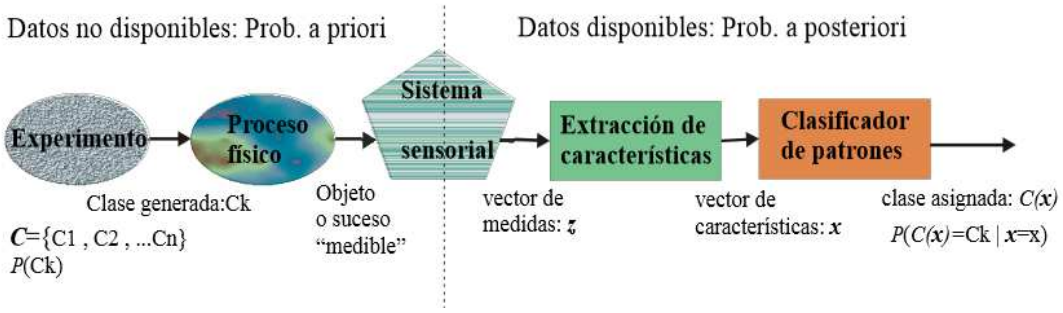

Figura 6.2.5 Estructura de un sistema clasificatorio [CID].

Se diferencian dos etapas bien definidas en el trabajo de investigación: la etapa de training o entrenamiento y la etapa de testing o clasificación.

La etapa de training consiste en introducir en el programa una serie de audios de los cuales se conoce su locutor original. Es necesario aplicar a estos audios el mismo procesado que se aplicará posteriormente a los audios en la etapa de testing para adecuarlos a las condiciones idóneas para trabajar (misma frecuencia de muestreo, reducción de ruido y eliminación de silencios). Una vez hecho esto, se procede a

extraer las características. Trabajaremos con coeficientes Cepstrum, una serie de coeficientes numéricos que aportan información sobre la señal basados en la percepción auditiva humana, derivados de la transformada de Fourier por tramas (o ventanas) de sonido (apartado 6.4). Cada audio de entrenamiento proporciona una matriz de coeficientes que se irán concatenando verticalmente hasta formar la matriz de entrenamiento, que será la que contenga los coeficientes cepstrales de todos los audios de train.

La etapa de clasificación es la que se encarga de analizar el audio desconocido para asociarlo finalmente a uno de los locutores con los que se ha entrenado el programa. El procesado es el mismo que en la anterior etapa, exceptuando que la matriz de clasificación sólo contendrá los coeficientes Cepstrum del audio a estudiar. Una vez se tienen las dos matrices, es importante normalizarlas para evitar futuros problemas por los distintos rangos de las variables. Si los rangos y varianzas son semejantes, el efecto será muy reducido pero si no es así, puede haber grandes desbalances.

El siguiente paso es utilizar un algoritmo que determine a qué locutor de entrenamiento se parece más cada trama de coeficientes del locutor a clasificar. De nuevo hay numerosos algoritmos para esta función; el que va a utilizarse en este caso es el algoritmo K-NN o de N vecinos cercanos. Este algoritmo asocia a cada trama de la matriz test su hablante más parecido basándose en los N vectores más similares de la matriz train. Para medir esa semejanza entre vectores de coeficientes se utiliza la distancia euclídea.

Tendremos por lo tanto un conjunto de valores que señalan con qué hablante se estima que se corresponde cada una de las tramas de sonido. Por último, sólo queda determinar a qué hablante se corresponde todo el audio, para lo cual bastará con elegir el locutor que más veces se haya estimado en el paso anterior.

El esquema de la figura 6.2.2 muestra el algoritmo seguido durante todo el programa hasta determinar a qué hablante pertenece cada audio de test y en los siguientes apartados se hablará en profundidad sobre cada uno de los bloques.

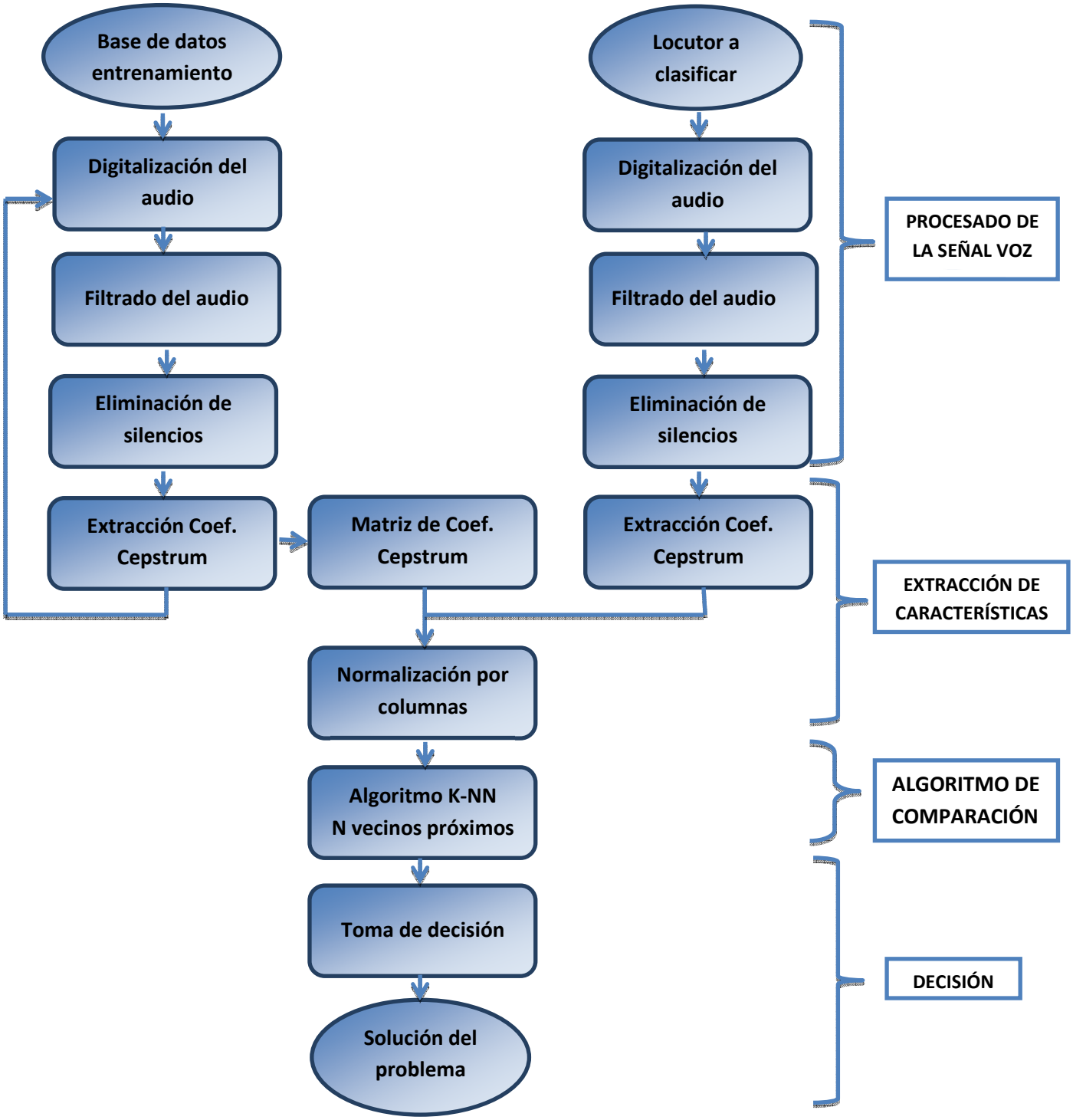

Figura 6.2.6 Esquema del algoritmo del sistema

3.3 Procesado de la señal de voz

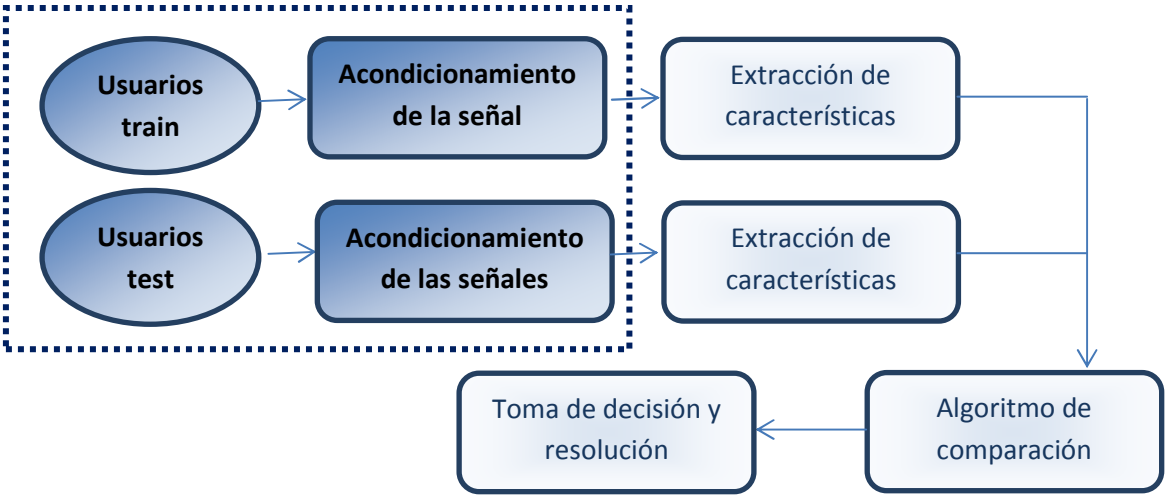

Figura 6.6.7 Procesado. Esquema del algoritmo.

El primer bloque del sistema consiste en la selección de los audios y un procesado de la voz que entra al sistema, con el objetivo de extraer sólo la información acústica relevante de los audios. Las señales de entrada pueden estar contaminadas con un ruido de fondo o con largos períodos sin habla que convenientemente deben ser eliminados. Esta función está dividida en tres pasos fundamentales: Selección de audios, reducción de ruido y supresión de silencios.

3.3.1 Selección de audios

Aparentemente puede parecer que la manera correcta de actuar es grabar a los distintos locutores que van a ser analizados en las mismas condiciones, bajo el mismo nivel de ruido y con el mismo equipo de grabación. Si bien es cierto que esto proporcionaría a priori un mejor resultado, la realidad es que a la hora de implantar estos sistemas en dispositivos comerciales, el usuario final no se va a encontrar siempre en las mismas condiciones acústicas, por lo que no daremos mucha importancia a este factor y se utilizarán grabaciones de voz de personajes públicos obtenidos de una página de contenido audiovisual de internet [YOU].

El habla se encuentra en un intervalo de bajas frecuencias comprendidas entre los 70Hz y los 1100Hz. Las cuantificaciones más usadas son las de 8 y 16 bits, mientras que los audios utilizados utilizan 62 bits, por defecto en la descarga de los mismos. La frecuencia de muestreo es de 22050Hz.

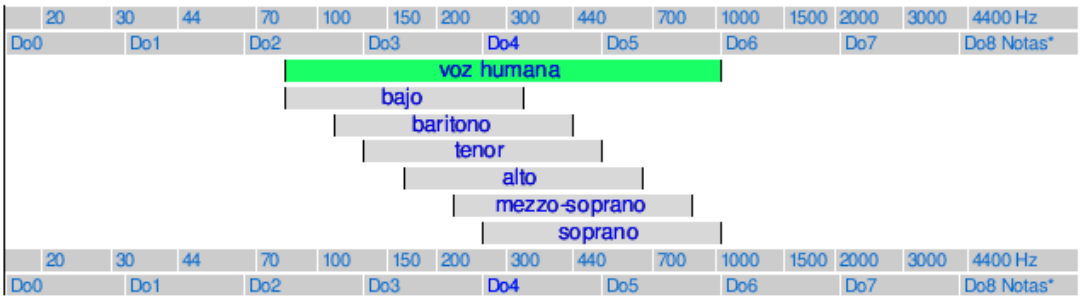

Figura6.6.1.8 Frecuencias de la voz humana [WIK].

Se han separado los audios en dos categorías:

- **Audios Train**: Son los audios con los que se entrena el programa. Este conjunto está formado por un audio de cada persona.

- **Audios test**: Son los audios que se quieren clasificar. Para llegar a una tabla de resultados se han utilizado cuatro audios diferentes de cada persona.

Los audios de entrenamiento tienen una duración de entre cinco y ocho segundos, aunque al eliminar los silencios posteriormente pasarán a tener una duración menor.

Los audios de clasificación, sin embargo, tienen duraciones muy dispares entre dos y nueve segundos, ya que interesa que el programa haga una buena diferenciación de locutores aunque la frase que diga la persona sea muy corta.

3.3.2 Reducción de ruido

Para la reducir los efectos indeseados del ruido que la señal tiene asociado se realiza un filtrado de la señal. Se ha optado por un paso banda de Butterworth que determinará la discriminación de voces espectralmente similares. Al filtrar la señal se

reducen las frecuencias indeseadas como ruido de fondo, música si la hay u otros ruidos que no son necesarios para el análisis.

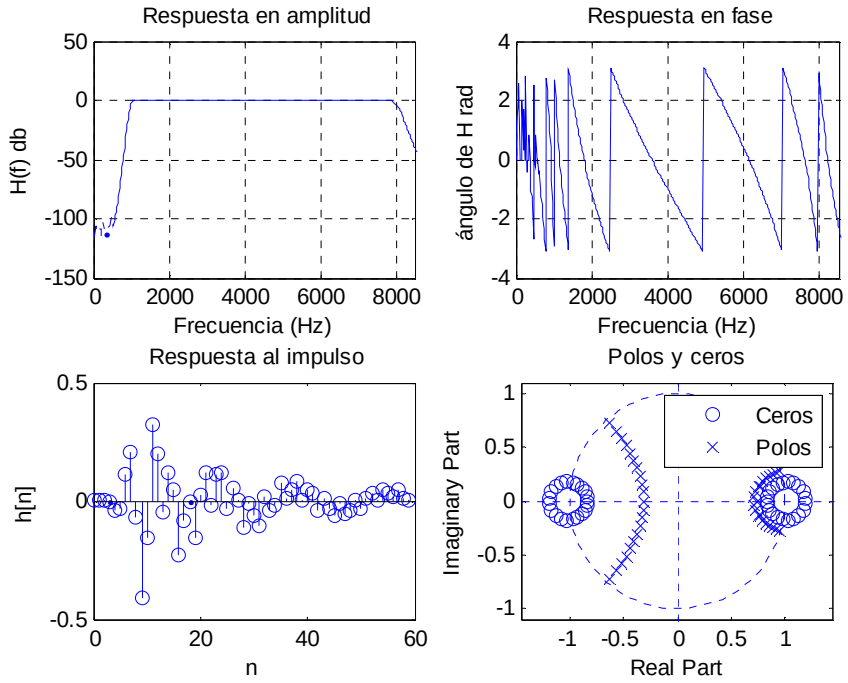

Figura 6.6.2.9 Filtro reductor de ruido.

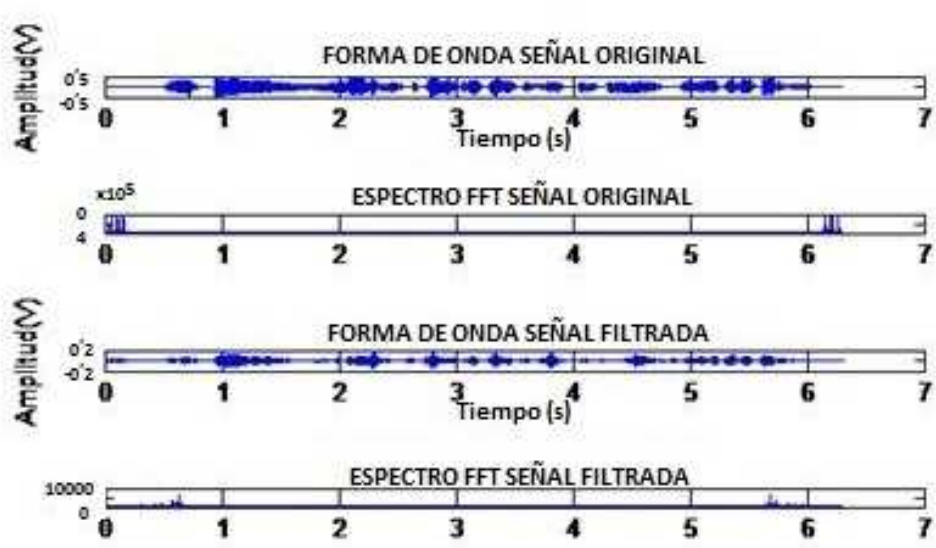

Figura 6.6.2.10 Espectro de las señales de entrada.

3.3.3 Supresión de silencios

Los silencios o períodos sin habla no aportan nada a un sistema de reconocimiento de locutor, por lo que la mejor opción es eliminarlos para que no influyan negativamente en la decisión final, además de disminuir los cálculos del programa.

En primer lugar se normaliza la señal respecto de su mayor nivel absoluto para independizar la forma de onda respecto de la intensidad de la señal. La señal normalizada se eleva al cuadrado y se divide por el número de muestras de la señal para obtener la energía promedio de la señal.

$$E_m = \sum_{i=0}^{n} \frac{x_i^2}{l}$$

Ecuación 6.6.6.1 Energía promedio de la señal.

A continuación se eliminan todos los períodos de silencio aplicando un umbral o threshold que elimina todos los segmentos de la señal por debajo del 10% de la energía media.

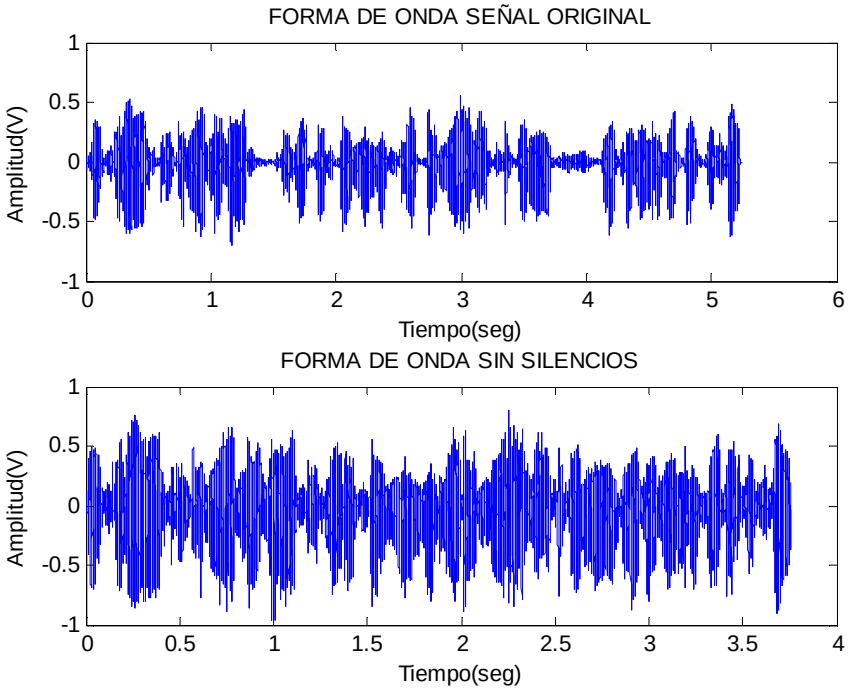

Figura 6.6.6.11 Espectro señal original y señal normalizada sin silencios.

3.4 Extracción de características

Figura 6.4.12 Extracción de características. Esquema del algoritmo.

La extracción de parámetros o características obtiene vectores de un determinado objeto que modelan su patrón. Para la extracción de características se ha optado por la técnica de Mel-Frequency Cepstral Coefficients. Como hemos descrito en capítulos anteriores, la voz puede simularse como un filtro vocal del cual puede representarse un espectro de las componentes frecuenciales de la voz y extraer información representativa. Los coeficientes Cepstrales son una representación del cepstrum de una señal ventaneada en el tiempo derivada de una Transformada Rápida de Fourier en una escala de frecuencias no lineal.

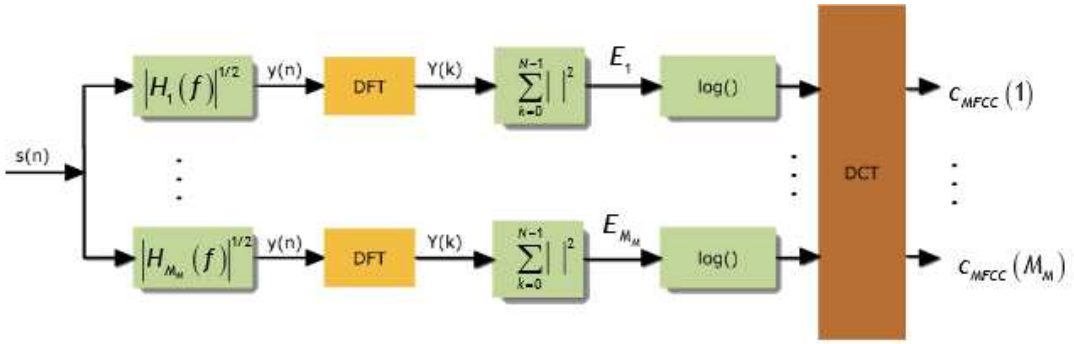

Figura 6.4.2 Mel-Frecuency Cepstral Coefficients [HTK].

En primer lugar se separa el audio en tramas para asumir que en cada trama la señal puede considerarse estacionaria. A continuación se realiza un enventanado Hamming a cada trama, que se utiliza para evitar las posibles discontinuidades que puedan aparecer al estar analizando sólo una fracción de audio o las componentes de alta frecuencia que pueden aparecer al introducir muestras con valor cero.

R. Hamming observó las respuestas de las ventanas utilizadas hasta entonces, y detectó que en la ventana uniforme y en la ventana de Von Hann los lóbulos latearles generalmente tenían signos opuestos. Diseñó otra ventana combinando las dos anteriores a la que puso su nombre, consiguiendo que la amplitud de los lóbulos laterales se redujera dando mejores resultados.

$$v(n) = a_0 - a_1 \cos\left(\frac{2\pi n}{N-1}\right) \text{ Donde } 0 \leq n \leq N\text{-}1$$

Ecuación 6.4.2 Ventana Hamming.

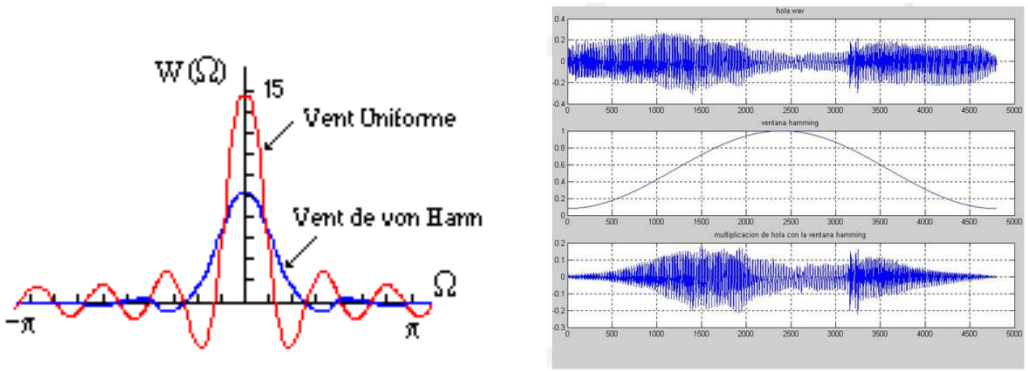

Figura 6.4.16 Combinación ventanas uniforme y Von Hann y efecto de ventana Hamming [TUT].

Se podría denominar coeficiente cepstral al principio de ponderar la energía que aporta a cada banda de frecuencias la señal bajo análisis y luego calcular en términos de un coeficiente para cada valor de energía en banda de frecuencia [SEC]. Para el cálculo de estos coeficientes se lleva a cabo el siguiente procedimiento:

A la señal resultante se le añaden ceros hasta conseguir tramas de 256 muestras sobre las que se aplica la Transformada Rápida de Fourier (FFT) para obtener el espectro de la señal. Seguidamente se le aplica un banco de filtros que permite la selección de bandas de frecuencia y que simulan la respuesta de la membrana basilar del oído humano.

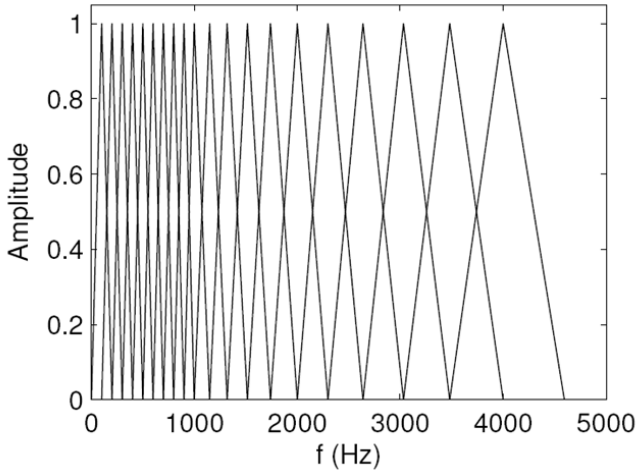

Figura 6.4.14 Banco de filtros MEL.

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{\frac{-j2\pi nk}{N}}$$ Donde k= 0,1,2… N-1

Ecuación 6.6.6 Transformada rápida de Fourier.

Los puntos superiores de cada función espectral f(m) se encuentran uniformemente espaciados en la escala de Mel en función de m y se determinan mediante la siguiente ecuación.

$$f(m) = \beta^{-1}\left(\beta(fi) + m\frac{\beta(fi) - \beta(fh)}{M+1}\right)$$

Ecuación 6.4.7 Puntos superiores de cada función espectral.

Se define *fi* como la frecuencia inferior y *fh* como la frecuencia superior del banco de filtros en Hz. *M* es el número de filtros, *m* el número de coeficientes a calcular y N el tamaño de la FFT.

Los filtros (Figura 6.4.5) son repartidos en el rango de frecuencias completas desde cero hasta la frecuencia de Nyquist, aunque en ocasiones se utiliza un criterio de limitación en banda para rechazar frecuencias no deseadas. Están espaciados linealmente para frecuencias menores a 1000Hz y logarítmicamente para frecuencias mayores de 1000Hz con el fin de capturar las características fonéticamente relevantes del habla.

Finalmente se calculan los coeficientes cepstrales, calculando primero el logaritmo de la energía de la salida de cada filtro (Ecuación 6.4.8) y posteriormente aplicando la Transformada Inversa de Fourier. Teniendo en cuenta que el logaritmo de la respuesta del filtro Mel es real y simétrico, la Transformada inversa de Fourier se reduce a la Transformada discreta del coseno (Ecuación 6.4.9).

$$S(m)) = \ln\left(\sum_{k=0}^{\frac{N}{2}-1} |X(k)|^2 H_m(k)\right), \quad 0 < m < M$$

Ecuación 6 5.8 Logaritmo de la energía a la salidas del filtro.

$$c(n) = \sum_{m=0}^{M-1} S(m) cos\left(\pi n\left(\frac{m-\frac{1}{2}}{M}\right)\right), \quad 0 \leq n \leq N-1$$

Ecuación 6.6.9 n-ésimo-N coeficiente cepstrum.

3.5 Algoritmo de comparación

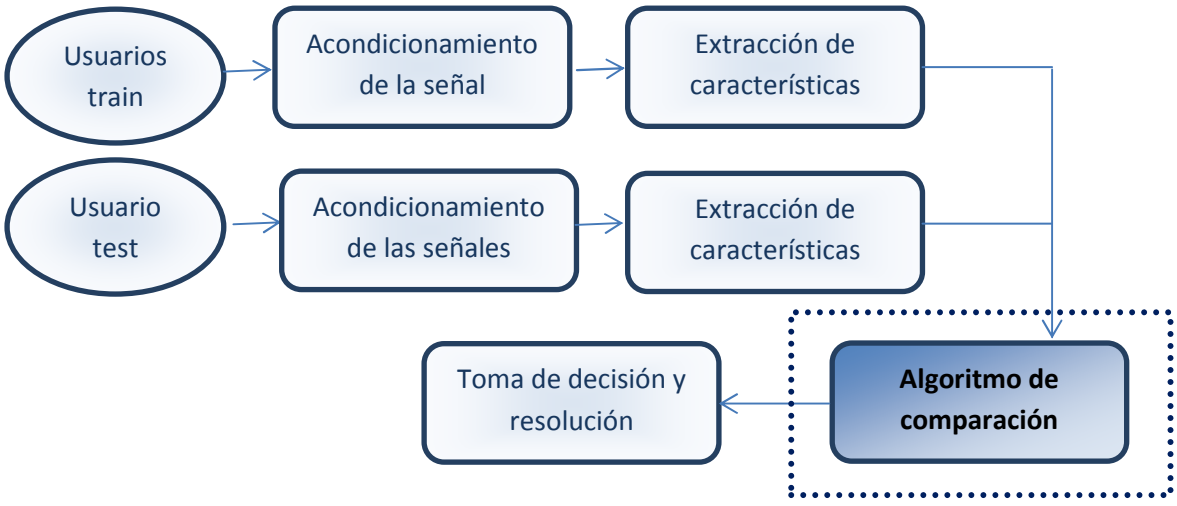

Figura 6.5.15 Algoritmo de clasificación. Esquema del algoritmo.

Para asignar una clase a un conjunto de parámetros se utiliza la clasificación de patrones. La asignación de clases se realiza para llevar a cabo una diferenciación entre subconjuntos de características [REC].

Para los sistemas de reconocimiento automático es necesario que los patrones que describen objetos de una misma clase presenten características similares. Hay distintos tipos de patrones:

- **Patrones vectoriales:** Codifican variables concretas significativas. Por ejemplo, la longitud de un pétalo. $\mathbf{X}= (x_1, x_2 \dots x_N)$
- **Patrones estructurados**: Codifican relaciones entre componentes del objeto o descriptores. Hay muchos tipos como por ejemplo árboles o cadenas. Un ejemplo sería el reconocimiento de huellas dactilares.

Los resultados pueden variar de forma notable utilizando un algoritmo u otro (Figura 6.5.1). Uno de los métodos que se utiliza para este propósito con patrones vectoriales es el algoritmo no paramétrico K-Means. Este algoritmo consigue la clasificación estudiando la métrica de distorsión entre vectores.

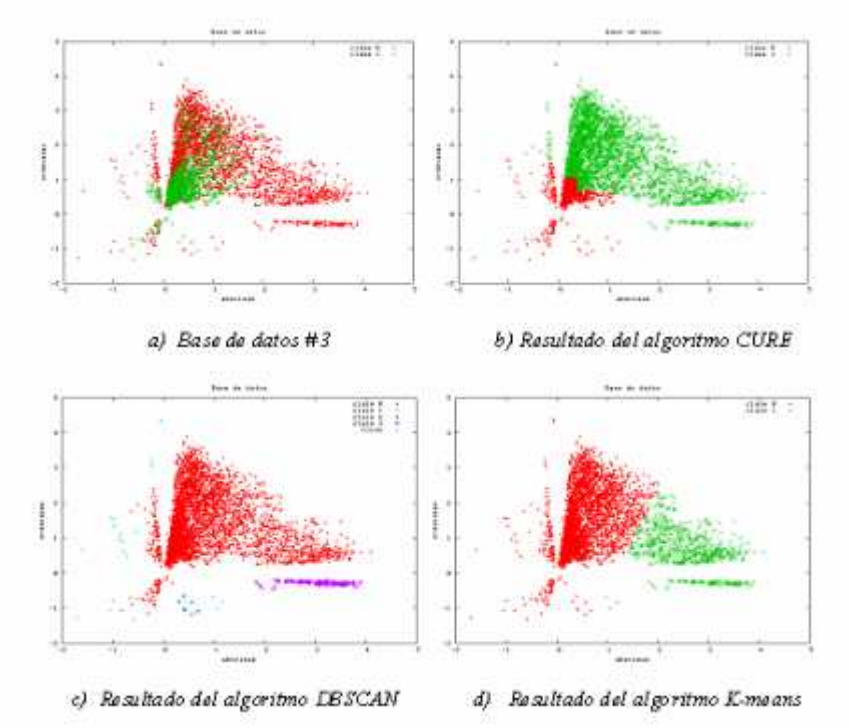

Figura 6.5.16 Diferentes algoritmos de clasificación [PAS].

3.5.1 Distancia Euclídea

La medida de distorsión más utilizada es la distancia Euclídea. Esta distancia se utiliza para el reconocimiento del hablante como método para calcular las diferencias existentes entre características. El resultado final de dicha comparación son valores numéricos que representan la distancia entre vectores de iguales dimensiones. La fórmula matemática que se emplea para el cálculo de la distancia Euclídea es la siguiente:

$$D = \sqrt{\sum_{i=1}^{N} \left(f_i - f_i^{`}\right)^2}$$

Ecuación 6.5.1.7 Distancia euclídea.

3.5.2 Algoritmo K-NN

La agrupación de objetos atendiendo a sus características ha sido ampliamente estudiada debido a sus numerosas aplicaciones como aprendizaje de máquina, minería de datos y descubrimiento de conocimiento.

El objetivo es reorganizar un grupo de objetos, en este caso pequeñas tramas de audio, los cuales tienen asociados vectores multidimensionales en grupos homogéneos, tales que los patrones de cada grupo son similares.

La idea es estimar la función de probabilidad en un punto x a partir de un conjunto de muestras. Se mide la distancia entre x y el punto o los puntos más próximos a x, es decir, sus vecinos más cercanos (Ecuación 6.5.2.1).

$$d_{NN}(x, R) = \min_{r_j \in R} \left| x - r_j \right|$$

Ecuación 6.5.2.8 Distancia del algoritmo K-NN.

En nuestro caso concreto, se miden las distancias euclídeas entre cada vector de coeficientes cepstrum del audio a clasificar y los vectores de la matriz de

entrenamiento, asumiendo así que los sonidos que tengan menor distancia entre ellos habrán sido producidos por la misma persona.

3.6 Toma de decisión

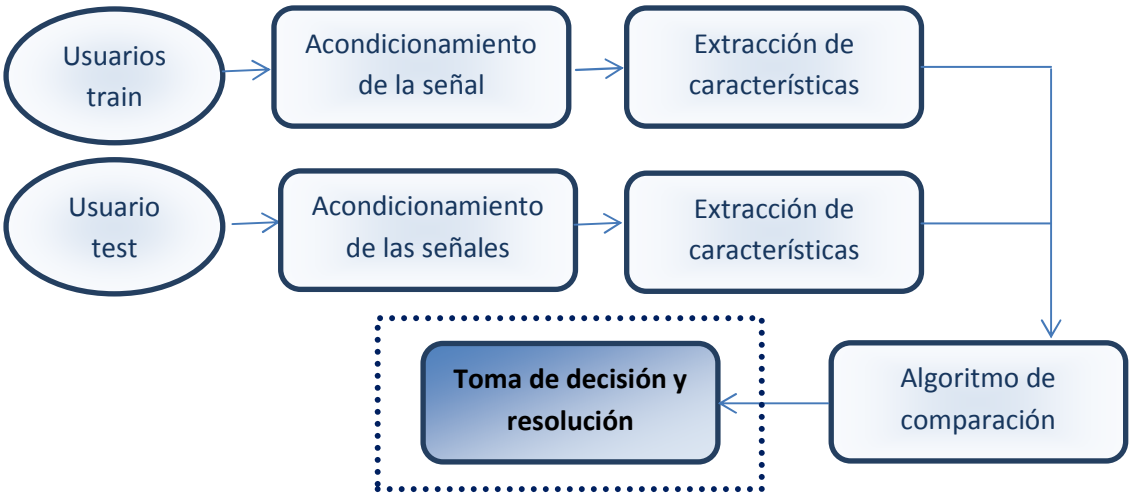

Figura 6.6.17 Toma de decisión. Esquema del algoritmo.

El algoritmo de comparación hace prácticamente todo el trabajo con respecto a la determinación de a qué clase pertenece cada objeto. En el caso estudiado lo que cataloga el algoritmo son fracciones de audios y no el audio al completo, por lo que el paso final es llegar a una conclusión para catalogar dicho audio.

Hemos elegido un detector simple consistente en elegir como hablante final aquel que se ha correspondido más veces con las fracciones de sonido estudiadas.

Por ejemplo, si el audio estudiado tiene 400 tramas y tras todo el proceso y aplicando el algoritmo K-NN se obtiene que 70 tramas se relacionan con el hablante 1, 600 tramas con el hablante 2 y 60 tramas con el hablante 6, se determinará como solución final al problema que el audio pertenece al hablante 2.

3.7 Implementación de código

Para realizar todo el trabajo se ha optado por utilizar el programa de software matemático Matlab (*MATrixLABoratoy*) que ofrece un entorno de desarrollo integrado con lenguaje propio de programación. El lenguaje de alto nivel Matlab permite una programación matemática mediante un entorno interactivo. Está indicado para el cálculo numérico, el análisis de datos y el desarrollo de algoritmos. El propio programa contiene librerías para tratamiento de audio que facilitan la lectura de los audios.

La función *wavread* de Matlab es capaz de leer un archivo de audio en formato *.wav* y generar automáticamente una matriz con los valores digitales del mismo, además de almacenar como variable su frecuencia de muestreo. Es importante darse cuenta de que los audios están almacenados en estéreo, es decir, contienen canal derecho e izquierdo (R/L) que en ámbitos musicales pueden distar bastante el uno del otro, pero que en el caso de la voz en los audios seleccionados son iguales a efectos prácticos, por lo que trabajaremos sólo con uno de ellos en modo mono. El comando *wavplay* permite escuchar la señal, siendo de gran ayuda para saber si se están haciendo las modificaciones correctamente y poder hacer un seguimiento de la señal en todo momento.

A la hora de realizar un trabajo de investigación, es importante que la estructura del mismo sea sencilla y fácil de modificar para poder hacer distintos experimentos sin mucho esfuerzo. Por ello en lugar de leer los audios directamente desde el programa, se han creado dos archivos de texto 'AudiosTrain.txt' y 'AudiosTest.txt' donde se puede modificar rápidamente los audios que queremos utilizar en cada iteración del programa.

El sistema implementado consta de once funciones, diez de ellas programadas para el propósito de este trabajo y una undécima (*melcepst.m*) que calcula los coeficientes Cepstrum obtenida de una librería de código ya existente [REP].

train.m
procesadoAudio.m
melcepst.m
eliminaSilencios.m
test.m
KNN.m
normalizaMatriz.m
distEuclidea.m
posMin.m
clasificador.m
valorMasRepetido.m

Tabla 6.7.2 Conjunto de funciones

El hilo principal del programa lo marca la función ***test.m*** la cual pone en marcha todo el programa llamando a las distintas funciones y recoge los audios a clasificar para su procesamiento. Un bucle leerá cada audio uno por uno y realizará la clasificación de forma independiente.

En primer lugar la función ***train.m*** recoge los audios de entrenamiento y los envía a ***procesadoAudio.m,*** que se encarga de filtrarlos y quitarles los silencios (***eliminaSilencios.m***) devolviendo limpias las señales a tratar. Una vez hecho esto, se realiza la extracción de características con ***melcepst.m*** formando una matriz de entrenamiento que contendrá todos los coeficientes Cepstrum de los audios de entrenamiento.

Este proceso sólo se realiza una vez en la ejecución del programa ya que los audios de entrenamiento no varían durante el proceso y son independientes del audio que se esté clasificando.

El audio de testing también se somete al procesado inicial haciendo uso de las mismas funciones, formando finalmente la matriz de clasificación, esta vez solo almacenando los coeficientes cepstrales de este audio.

Una vez se tienen las dos matrices, se realiza la normalización con ***normalizaMatriz.m.*** Éste método devuelve las dos matrices normalizadas por columnas dividiendo entre el máximo de la columna train y test.

El siguiente paso es mandar las dos matrices a ***kNN.m*** junto con el número de vecinos cercanos que se quiere analizar. El algoritmo de clasificación se ayuda de ***disEuclidea.m*** para obtener la distancia euclídea entre vectores y de ***posMin.m*** para encontrar las posiciones de los x vectores con distancia más pequeña.

Estos vectores con menor distancia se envían a ***clasificador.m*** que asigna a cada trama el supuesto hablante con el que se estima que se corresponde por similitud. Por último se envía el vector con los supuestos hablantes a ***valorMasRepetido.m*** que busca en el vector el hablante con el que más veces se ha relacionado el audio y lo devuelve determinando que esa es la solución al problema planteado.

Capítulo 4: RESULTADOS Y EVALUACIÓN

En total se han utilizado setenta y cinco audios distintos de quince personas diferentes (cinco audios por persona), entre ellos hombres y mujeres en tres idiomas diferentes: castellano, árabe e inglés.

Locutor	Nombre	Sexo	Idioma
Locutor 1:	Esperanza Aguirre y Gil de Biedma	M	Español
Locutora 2:	Marta Sánchez López	M	Español
Locutor 6:	Franklin Tshimini Nsombolay (FrankT)	H	Español
Locutor 4:	José Ignacio Gabilondo Pujol	H	Español
Locutor 5:	Rafael Nadal Parera	H	Español
Locutor 6:	Jordi Évole Requena	H	Español
Locutor 7:	Arturo Pérez-Reverte Gutiérrez	H	Español
Locutora 8:	Rosa María Díez González	M	Español
Locutor 9:	Willard Christopher *Smith* Jr. (Will Smith)	H	Inglés
Locutora 10:	María Nieves Rebolledo Vila (Bebe)	M	Español
Locutor 11:	Aboubakr Jamaï	H	Árabe
Locutor 12:	Abdelilá Benkirane	H	Árabe
Locutora 16:	Wafa Sultán	M	Árabe
Locutora 14:	Anna Torv	M	Inglés
Locutor 15:	Jhon Noble	H	Inglés

Tabla 4.6 Relación de locutores, sexo e idioma.

La duración media de los audios es de 4´46 s. Un audio de cada locutor se utiliza para entrenar al programa y los otros cuatro restantes se someten a clasificación. El número de audios utilizados en cada momento dependerá del experimento que se esté llevando a cabo.

Trabajaremos sobre tres parámetros variables que se irán modificando para encontrar una combinación lo más acertada posible para la resolución del problema, que puede no ser única en todos los casos. Estas variables serán el número de coeficientes cepstrum por trama, el número de filtros y los K vecinos más cercanos del algoritmo K-NN.

Para la primera variable, el número de coeficientes, se ha optado por los valores 10, 15, 18, 24, 28 y 62. Estos valores se han elegido ya que son los valores típicos utilizados para sistemas de reconocimiento de voz.

Para la segunda variable se han elegido 10, 20, 60, 40, 50 ó 60 filtros ya que no tendría sentido dividir el espectro frecuencial en un número mayor.

El número de vecinos cercanos ha sido elegido evitando números pares para que no se produzca empate al comparar el vector estudiado con los de entrenamiento. La siguiente figura muestra un ejemplo donde se ve claramente la importancia de escoger un valor impar. Suponiendo que la estrella es el vector a comparar, con k=6 se concluye que pertenece al grupo de círculos azules y con k=5 que pertenece al grupo de rectángulos verdes, mientras que si elegimos k=4, no se podría dar una solución.

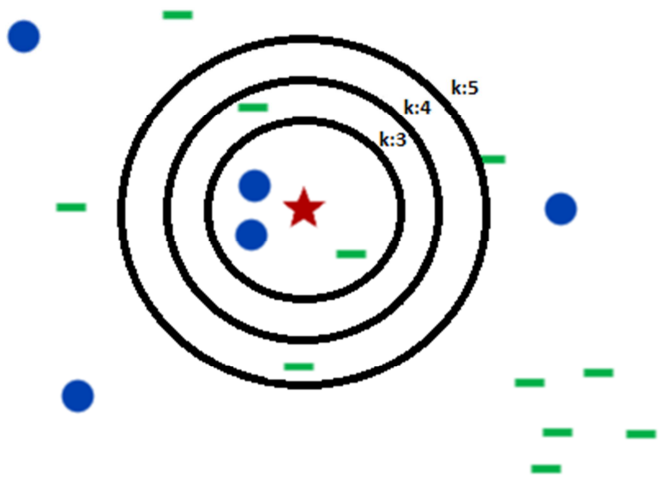

Figura 4.18 Algoritmo K-NN.

4.1 Experimento #1: Dos hablantes de distinto sexo

El primer experimento trata de diferenciar dos locutores de sexo opuesto: masculino y femenino. Generalizando se puede decir que las frecuencias que aborda la voz masculina se encuentran entre los 70Hz y los 200Hz, mientras que la voz femenina tiende a ser más aguda encontrándose entre los 150Hz y los 650Hz.

Este experimento se repetirá con dos parejas de audio mujer-hombre diferentes.

Se ejecutará el programa 72 veces con distintas combinaciones de parámetros tomando los siguientes valores:

- Coeficientes Cepstrum: 10, 18, 24 ó 62.
- Número de filtros: 10, 20, 60, 40, 50 ó 60.
- Valor k de K-NN: 1, 6 ó 5.

4.1.1 Experimento #1.1

En el primer caso se utilizan los audios pertenecientes a los hablantes 4 y 10 (hombre y mujer). Este caso implica dos audios de entrenamiento y ocho audios a clasificar. Los resultados obtenidos se muestran en las siguientes tablas. *Nc* se corresponde con el número de coeficientes Cepstrum por trama y *p* con el número de filtros. Los valores resultantes son el porcentaje de acierto obtenido normalizado de 0 a 1.

Experimento #1.1															
Nc,p	**K=1**	**K=6**	**K=5**	**Nc,p**	**K=1**	**K=6**	**K=5**	**Nc,p**	**K=1**	**K=6**	**K=5**	**Nc,p**	**K=1**	**K=6**	**K=5**
10,10:	0.5	0.5	0.5	**18,10:**	0.5	0.5	0.5	**24,10:**	0.5	0.5	0.5	**62,10:**	0.5	0.5	0.5
10,20:	**1**	**1**	**1**	**18,20:**	**1**	**1**	**1**	**24,20:**	0.5	0.5	0.5	**62,20:**	0.5	0.5	0.5
10,60:	**1**	**1**	**1**	**18,60:**	**1**	**1**	**1**	**24,60:**	**1**	**1**	**1**	**62,60:**	0.5	0.5	0.5
10,40:	**1**	**1**	**1**	**18,40:**	**1**	**1**	**1**	**24,40:**	**1**	**1**	**1**	**62,40:**	**1**	**1**	**1**
10,50:	**1**	**1**	**1**	**18,50:**	**1**	**1**	**1**	**24,50:**	**1**	**1**	**1**	**62,50:**	**1**	**1**	**1**
10,60:	**1**	**1**	**1**	**18,60:**	**1**	**1**	**1**	**24,60:**	**1**	**1**	**1**	**62,60:**	**1**	**1**	**1**

Tabla 4.4.1.1 Resultados experimento #1.1.

4.1.2 Experimento #1.2

En el segundo caso se repite la misma configuración para una pareja de audios diferente, los hablantes 1 y 7 (mujer y hombre), obteniendo los siguientes resultados.

Experimento #1.2															
Nc,p	**K=1**	**K=6**	**K=5**	**Nc,p**	**K=1**	**K=6**	**K=5**	**Nc,p**	**K=1**	**K=6**	**K=5**	**Nc,p**	**K=1**	**K=6**	**K=5**
10,10:	0.5	0.5	0.5	**18,10:**	0.5	0.5	0.5	**24,10:**	0.5	0.5	0.5	**62,10:**	0.5	0.5	0.5
10,20:	**1**	0.875	0.875	**18,20:**	**1**	0.875	0.875	**24,20:**	0.5	0.5	0.5	**62,20:**	0.5	0.5	0.5
10,60:	0.75	0.75	0.875	**18,60:**	0.75	0.75	0.875	**24,60:**	0.75	0.75	0.875	**62,60:**	0.5	0.5	0.5
10,40:	**1**	**1**	**1**	**18,40:**	**1**	**1**	**1**	**24,40:**	**1**	**1**	**1**	**62,40:**	**1**	0.875	**1**
10,50:	**1**	**1**	**1**	**18,50:**	**1**	**1**	**1**	**24,50:**	**1**	**1**	0.875	**62,50:**	**1**	**1**	**1**
10,60:	**1**	**1**	**1**	**18,60:**	**1**	**1**	**1**	**24,60:**	**1**	**1**	**1**	**62,60:**	**1**	**1**	**1**

Tabla 4.5.1.2 Resultados experimento #1.2.

Observando el alto porcentaje de combinaciones donde el programa es capaz de reconocer todos los audios introducidos a la perfección, se puede afirmar que para parejas de sexo opuesto el sistema es adecuado. Parece conveniente utilizar valores bajos de coeficientes y de vecinos cercanos ya que el programa realiza menor número de operaciones reduciendo el coste computacional y el tiempo de ejecución. En cuanto al número de filtros, se necesitan al menos 40 para acertar todos los audios.

4.2 Experimento #2: Dos hablantes del mismo sexo

Diferenciar dos personas del mismo sexo tiene la complicación añadida de que los rangos frecuenciales pueden ser muy similares. El segundo experimento se divide de nuevo en dos, hombre-hombre y mujer-mujer. Se ejecutará el programa con 72 combinaciones de parámetros tomando los siguientes valores:

- Coeficientes Cepstrum: 10, 18, 24 ó 62.
- Número de filtros: 10, 20, 60, 40, 50 ó 60.
- Valor k de K-NN: 1, 6 ó 5.

4.2.1 Experimento #2.1

Se utilizan audios de dos voces masculinas, los hablantes 4 y 6. Los resultados son los siguientes.

Experimento #2.1															
Nc,p	**K=1**	**K=6**	**K=5**	**Nc,p**	**K=1**	**K=6**	**K=5**	**Nc,p**	**K=1**	**K=6**	**K=5**	**Nc,p**	**K=1**	**K=6**	**K=5**
10,10:	0.5	0.5	0.5	**18,10:**	0.5	0.5	0.5	**24,10:**	0.5	0.5	0.5	**62,10:**	0.5	0.5	0.5
10,20:	1	1	1	**18,20:**	1	1	1	**24,20:**	0.5	0.5	0.5	**62,20:**	0.5	0.5	0.5
10,60:	1	1	1	**18,60:**	1	1	1	**24,60:**	1	1	1	**62,60:**	0.5	0.5	0.5
10,40:	1	1	1	**18,40:**	1	1	1	**24,40:**	1	1	1	**62,40:**	1	1	1
10,50:	1	1	1	**18,50:**	1	1	1	**24,50:**	1	1	1	**62,50:**	1	1	1
10,60:	1	1	1	**18,60:**	1	1	1	**24,60:**	1	1	1	**62,60:**	1	1	1

Tabla 4.2.6.1 Resultados experimento #2.1.

4.2.2 Experimento #2.2

Dos voces femeninas correspondientes a los audios 2 y 10.

Experimento #2.2															
Nc,p	**K=1**	**K=6**	**K=5**	**Nc,p**	**K=1**	**K=6**	**K=5**	**Nc,p**	**K=1**	**K=6**	**K=5**	**Nc,p**	**K=1**	**K=6**	**K=5**
10,10:	0.5	0.5	0.5	**18,10:**	0.5	0.5	0.5	**24,10:**	0.5	0.5	0.5	**62,10:**	0.5	0.5	0.5
10,20:	0.875	0.875	0.75	**18,20:**	1	1	1	**24,20:**	0.5	0.5	0.5	**62,20:**	0.5	0.5	0.5
10,60:	1	0.75	0.875	**18,60:**	1	1	1	**24,60:**	1	1	1	**62,60:**	0.5	0.5	0.5
10,40:	1	1	0.75	**18,40:**	1	1	1	**24,40:**	1	1	1	**62,40:**	1	1	1
10,50:	1	0.875	0.75	**18,50:**	1	1	1	**24,50:**	1	1	1	**62,50:**	1	1	1
10,60:	1	0.875	0.75	**18,60:**	1	1	1	**24,60:**	1	1	1	**62,60:**	1	1	1

Tabla 4.2.7.2 Resultados experimento #2.2.

Del experimento #2, diferenciación entre mujer-mujer u hombre-hombre, se puede afirmar que el sistema también da un resultado óptimo, reconociendo un 100% de las veces el locutor analizado con muchas de las combinaciones de parámetros. Se observa esta vez que con un número bajo de coeficientes el programa tiene más dificultad para diferenciar entre mujeres, necesitando al menos vectores de 18 coeficientes Cepstrum.

4.3 Experimento #3: Distintos idiomas

Algunas lenguas tienen fonemas diferentes que incluyen sonidos aspirados o nasales. El tercer experimento trata de probar el sistema con dos idiomas distintos al castellano.

Se ejecutará el programa con 72 combinaciones de parámetros tomando los siguientes valores:

- Coeficientes Cepstrum: 10, 18, 24 ó 62.
- Número de filtros: 10, 20, 60, 40, 50 ó 60.
- Valor k de K-NN: 1, 6 ó 5.

4.3.1 Experimento #3.1

El árabe estándar moderno tiene una pronunciación muy diferente al castellano entre otras razones porque sólo tiene tres vocales: /a/, /i/, /u/ y dos diptongos: /aj/ y /aw/. Para este experimento se utilizarán tres locutores de habla árabe, dos hombres y una mujer correspondientes a los audios 11, 12 y 16.

Experimento #6.1															
Nc,p	**K=1**	**K=6**	**K=5**	**Nc,p**	**K=1**	**K=6**	**K=5**	**Nc,p**	**K=1**	**K=6**	**K=5**	**Nc,p**	**K=1**	**K=6**	**K=5**
10,10:	0.66	0.66	0.66	**18,10:**	0.66	0.66	0.66	**24,10:**	0.66	0.66	0.66	**62,10:**	0.66	0.66	0.66
10,20:	0.86	0.66	0.66	**18,20:**	0.75	0.66	0.66	**24,20:**	0.66	0.66	0.66	**62,20:**	0.66	0.66	0.66
10,60:	0.92	0.92	0.75	**18,60:**	0.92	0.86	0.66	**24,60:**	0.92	0.75	0.66	**62,60:**	0.66	0.66	0.66
10,40:	0.92	0.92	0.75	**18,40:**	0.86	0.75	0.75	**24,40:**	0.92	0.86	0.66	**62,40:**	0.92	0.66	0.66
10,50:	**1**	0.86	0.75	**18,50:**	0.92	0.86	0.75	**24,50:**	0.86	0.86	0.75	**62,50:**	0.92	0.86	0.66
10,60:	**1**	0.92	0.75	**18,60:**	0.92	0.86	0.75	**24,60:**	0.92	0.92	0.75	**62,60:**	0.92	0.86	0.66

Tabla 4.6.1.8 Resultados experimento #6.1.

4.3.2 Experimento #3.2

La lengua inglesa utiliza los mismos símbolos que el castellano pero tiene fonemas muy diferentes. Se utilizarán tres voces de personas hablando en inglés, que serán los audios 9, 14 y 15.

Experimento #6.2															
Nc,p	**K=1**	**K=6**	**K=5**	**Nc,p**	**K=1**	**K=6**	**K=5**	**Nc,p**	**K=1**	**K=6**	**K=5**	**Nc,p**	**K=1**	**K=6**	**K=5**
10,10:	0.66	0.66	0.66	**18,10:**	0.66	0.66	0.66	**24,10:**	0.66	0.66	0.66	**62,10:**	0.66	0.66	0.66
10,20:	0.75	0.66	0.66	**18,20:**	0.86	0.66	0.66	**24,20:**	0.66	0.66	0.66	**62,20:**	0.66	0.66	0.66
10,60:	0.86	0.66	0.66	**18,60:**	0.92	0.66	0.66	**24,60:**	0.92	0.66	0.66	**62,60:**	0.66	0.66	0.66
10,40:	0.86	0.75	0.66	**18,40:**	0.92	0.66	0.66	**24,40:**	0.92	0.66	0.75	**62,40:**	**1**	0.92	0.86
10,50:	0.86	0.75	0.66	**18,50:**	0.92	0.66	0.66	**24,50:**	0.92	0.66	0.66	**62,50:**	**1**	0.75	0.86
10,60:	0.86	0.75	0.66	**18,60:**	0.92	0.66	0.66	**24,60:**	0.92	0.66	0.75	**62,60:**	**1**	0.86	0.75

Tabla 4.6.9.2 Resultados experimento #6.2.

El tercer experimento ofrece datos diferentes en función del idioma. Si bien funcionan mejor con un vecino cercano y un número elevado de filtros, el árabe estándar tiene mayor probabilidad de acierto con pocos coeficientes espectrales (10) mientras que el inglés necesita una cifra elevada de filtros (50-60) para obtener un resultado del 100% de acierto.

Como conclusión de este apartado afirmaremos que si bien el algoritmo es válido para otros idiomas diferentes al castellano, sería necesario optimizar los parámetros y probar otras combinaciones diferentes para afinar el sistema a otras pronunciaciones.

4.4 Experimento #4: Conjunto de hablantes

El último experimento consiste en evaluar el funcionamiento del sistema al introducir el conjunto de locutores. Se toma un audio de entrenamiento por hablante y se utiliza el resto para la clasificación.

Se ejecutará el programa con 108 combinaciones de parámetros tomando los siguientes valores:

- Coeficientes Cepstrum: 10, 15, 18, 24, 28 ó 62.
- Número de filtros: 10, 20, 60, 40, 50 ó 60.
- Valor k de K-NN: 1, 6 ó 5.

Se ha elegido un mayor número de combinaciones ya que al aumentar notablemente el número de señales de audio de entrenamiento se prevé que se obtendrán resultados más dispares entre las probabilidades de acierto.

Nc,p	Experimento 4.a) Un vecino cercano										
10,10:	0.1	**15,10:**	0.1	**18,10:**	0.1	**24,10:**	0.1	**28,10:**	0.1	**62,10:**	0.1
10,20:	0.85	**15,20:**	0.9	**18,20:**	0.9	**24,20:**	0.1	**28,20:**	0.1	**62,20:**	0.1
10,60:	0.825	**15,60:**	**0.925**	**18,60:**	**0.925**	**24,60:**	0.875	**28,60:**	0.9	**62,60:**	0.1
10,40:	0.825	**15,40:**	0.875	**18,40:**	0.875	**24,40:**	**0.925**	**28,40:**	0.85	**62,40:**	0.875
10,50:	0.825	**15,50:**	0.825	**18,50:**	0.9	**24,50:**	0.85	**28,50:**	0.875	**62,50:**	0.875
10,60:	0.825	**15,60:**	0.9	**18,60:**	0.875	**24,60:**	0.85	**28,60:**	0.85	**62,60:**	0.875

Tabla 4.4.10 Experimento 4.a.

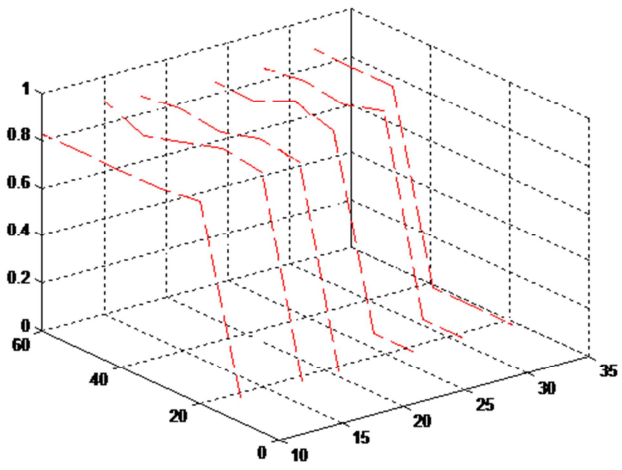

Figura 4.4.19 Resultados gráficos experimento 4.a.

Nc,p:	Experimento 4.b) Tres vecinos cercanos										
10,10:	0.1	**15,10:**	0.1	**18,10:**	0.1	**24,10:**	0.1	**28,10:**	0.1	**62,10:**	0.1
10,20:	0.475	**15,20:**	0.55	**18,20:**	**0.6**	**24,20:**	0.1	**28,20:**	0.1	**62,20:**	0.1
10,60:	0.5	**15,60:**	**0.6**	**18,60:**	**0.6**	**24,60:**	0.55	**28,60:**	**0.6**	**62,60:**	0.1
10,40:	0.5	**15,40:**	0.575	**18,40:**	0.525	**24,40:**	0.525	**28,40:**	0.575	**62,40:**	0.525
10,50:	0.525	**15,50:**	0.55	**18,50:**	0.575	**24,50:**	0.55	**28,50:**	0.525	**65,50:**	0.525
10,60:	0.65	**15,60:**	0.55	**18,60:**	0.55	**24,60:**	0.525	**28,60:**	0.525	**62,60:**	0.5

Tabla 4.4.11 Experimento 4.b.

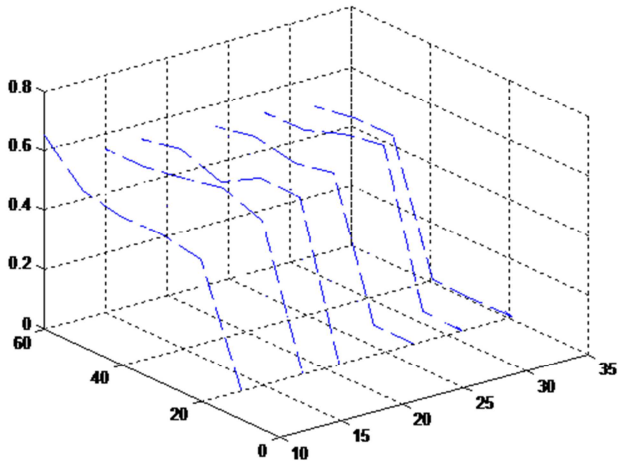

Figura 4.4.20 Resultados gráficos experimento 4.b.

Nc,p	Experimento 4.c) Cinco vecinos cercanos										
10,10:	0.1	**15,10:**	0.1	**18,10:**	0.1	**24,10:**	0.1	**28,10:**	0.1	**62,10:**	0.1
10,20:	0.6	**15,20:**	0.75	**18,20:**	0.75	**24,20:**	0.1	**28,20:**	0.1	**62,20:**	0.1
10,60:	0.675	**15,60:**	0.75	**18,60:**	**0.775**	**24,60:**	0.75	**28,60:**	0.725	**62,60:**	0.1
10,40:	0.7	**15,40:**	0.75	**18,40:**	0.75	**24,40:**	0.675	**28,40:**	0.75	**62,40:**	0.725
10,50:	0.75	**15,50:**	**0.775**	**18,50:**	**0.775**	**24,50:**	0.65	**28,50:**	0.7	**65,50:**	0.725
10,60:	0.775	**15,60:**	0.5	**18,60:**	0.8	**24,60:**	0.675	**28,60:**	0.675	**62,60:**	0.675

Tabla 4.4.12 Experimento 4.c.

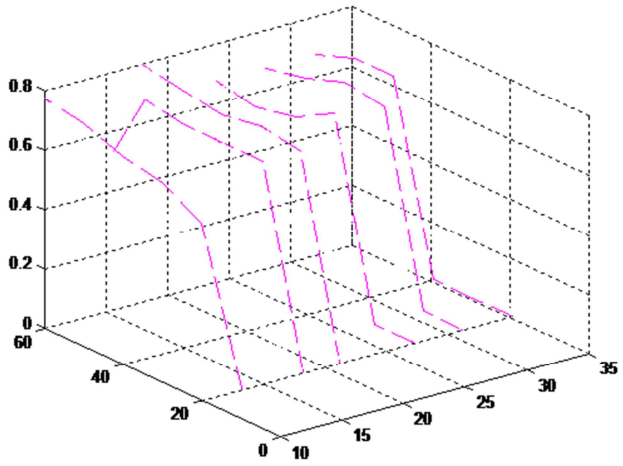

Figura 4.4.21 Resultados gráficos experimento 4.c.

El experimento #4 es el que aporta mayor información a efectos prácticos. La mayor probabilidad de acierto conseguida es del 92´5% de acierto, donde la mayoría de errores que provoca el algoritmo se producen entre voces femeninas con similar tono de voz. De nuevo utilizar un vecino cercano da mejores resultados. Este experimento reduce mucho el margen de combinaciones entre nº. coeficientes y nº. filtros optimizando el resultado en tres casos:

- Un vecino cercano. 15 coeficientes. 60 filtros.
- Un vecino cercano. 18 coeficientes. 60 filtros.
- Un vecino cercano. 24 coeficientes. 40 filtros.

4.5 Evaluación

Con toda la información anterior se puede hacer una valoración funcional del sistema en conjunto.

Por una parte, se necesitan al menos 20 filtros para obtener la información necesaria para implementar el sistema, con 10 el programa no es capaz de diferenciarlos y resuelve que todos son la misma persona. A mayor número de coeficientes Cepstrum el número de filtros necesarios es mayor.

Parece extraño que sólo con un vecino cercano el sistema dé mejores resultados que con tres o con cinco. La explicación que se muestra más acertada es que las tramas son de tan baja duración que pueden albergar la pronunciación de un solo fonema, de manera que al comparar un vector con el algoritmo de clasificación solo podría ser similar al fonema al que realmente se corresponde y al aceptar varios vecinos obtiene similitudes con fonemas diferentes que derivan en erratas.

En general, si el programa tuviera que implementarse con una sola configuración, los mejores resultados se obtendrían utilizando un vecino cercano, 24 coeficientes cepstrales y 40 filtros por trama, mientras que viendo las diferencias entre experimentos, lo mejor sería adaptar estos parámetros en función de los audios de entrenamiento cada vez que estos varíen buscando siempre el máximo acierto.

Capítulo 5: PLANIFICACIÓN Y PRESUPUESTO

5.1 Planificación

El proyecto de investigación se basa en la construcción de una herramienta software capaz de reconocer a un determinado locutor entre un conjunto de locutores con los que previamente se ha entrenado el programa.

Para su desarrollo se ha dividido el trabajo en diferentes etapas:

- **Recopilación de información y estado del arte**

  Se necesita un informe previo de los trabajos existentes y las referencias tecnológicas que se emplean en este tipo de proyectos. Es importante diferenciar los sistemas de reconocimiento de hablante de los sistemas de reconocimiento de voz.

- **Referencias de Matlab**

  Matlab tiene lenguaje de programación propio que incluye numerosas funciones de tratamiento de audio. Para trabajar con este software se necesita una familiarización con el entorno y sus funciones.

- **Análisis funcional**

  En esta etapa se analiza toda la información relativa al proyecto para formar la idea del esquema y el modelo que posteriormente se quiere implantar.

- **Obtención de audios e implementación**

  Se obtienen los audios necesarios para el sistema y se programa el algoritmo completo que realizará la clasificación.

- **Pruebas y experimentación**

  Se prueba en primer lugar que la implementación del código funcione y en segundo lugar se realiza el conjunto de experimentos necesarios para ajustar los parámetros y llegar a la solución requerida.

- **Documentación**

  Se documenta todo el proceso y se crea la memoria de proyecto.

En el siguiente diagrama de Gantt se muestra el esquema del tiempo dedicado a estas tareas:

Proyecto	Fecha Inicial	Fecha final	Jornadas
Recopilación información	09/09/2013	19/09/2013	
Estado del arte	20/09/2013	25/09/2013	
Referencias de Matlab	26/09/2013	01/10/2013	
Análisis funcional	02/10/2013	12/10/2013	
Obtención de audios	14/10/2013	17/10/2013	
Implementación Matlab	18/10/2013	12/12/2013	
Pruebas código	13/12/2013	20/12/2013	
Experimentos del sistema	23/12/2013	07/01/2014	
Documentación	08/01/2014	06/02/2014	

Figura 5.1.22 Diagrama de Gantt.

5.2 Presupuesto

En el presupuesto se recogen los recursos necesarios para el proyecto así como los costes directos e indirectos que implica la realización de un sistema de reconocimiento de hablante.

5.2.1. Recursos y roles

Se necesitan dos perfiles diferentes para el correcto desarrollo del trabajo:

- **Ingeniero Sénior (Tutor).** Ingeniero titulado experto en sistemas de reconocimiento por análisis de características. Es el encargado de supervisar el proyecto durante todo su recorrido así como de revisar la documentación del mismo.

- **Analista Investigadora (Alumna).** Posee el perfil práctico con conocimientos en programación en Matlab. Se encarga de revisar toda la documentación anterior al proyecto, estado del arte y las opciones de

mercado. Lleva a cabo el desarrollo del sistema y realiza las pruebas pertinentes a la aplicación.

5.2.2 Costes directos

En primer lugar se debe calcular la dedicación de los recursos al proyecto en unidad de jornadas. Para el cálculo del salario por jornada se ha tenido en cuenta una ponderación basada en tarifas de varias consultoras.

- Salario Ingeniero Sénior: 180€/jornada.
- Salario Analista Investigador: 90€/jornada.

Tarea	Dedicación total	Dedicación Ing. Sénior	Dedicación analista investigador.
Recopilación de información	**10**	**10%**	**90%**
Estado del arte	**5**	**0%**	**100%**
Referencias Matlab	**5**	**10%**	**90%**
Análisis funcional	**10**	**60%**	**70%**
Obtención de audios	**4**	**0%**	**100%**
Implementación Matlab	**40**	**0%**	**100%**
Pruebas de código	**6**	**0%**	**100%**
Experimentos del sistema	**10**	**0%**	**100%**
Documentación	**22**	**15%**	**85%**
TOTAL JORNADAS	**112**	**8**	**104**

Tabla 5.2.2.16 Cálculo de dedicación en jornadas.

Lo que hace un total de 11520€ en costes directos de recursos humanos.

En cuanto a material, al no utilizar un software de libre distribución tendremos los gastos recogidos en la siguiente tabla.

Descripción	A. Coste (€)	B. %Uso dedicado proyecto	C. Dedicación (meses)	D. Período depreciación (meses)	E. Coste imputable
Software de programación Matlab	69€	100%			69€
Equipo i6. 4Gb RAM.	450€	90%	6	60	40.5€

Tabla 5.2.2.14 Cálculo de material empleado.

El coste imputable se calcula mediante la amortización con la siguiente fórmula.

$$E = \frac{C}{D} * A * B$$

Ecuación 5.2.2.9 Fórmula de amortización.

Recopilando los datos anteriores, el total de costes directos es de **11.660€.**

5.2.3 Costes indirectos

En los costes indirectos se valoran los costes de gestión y seguimiento del proyecto, estimándose a priori como un 20% de los costes directos, lo que hace un total de **2.626€.**

5.2.4 Presupuesto total

Costes directos: 11.660€

Costes indirectos: 2.626€

Total presupuesto: 16.956€

CONCLUSIÓN Y TRABAJOS FUTUROS

El reconocimiento de hablante supone un gran avance en la tecnología que permite la comunicación entre seres humanos y máquinas. El estudio de estos sistemas puede mejorar la calidad de muchos dispositivos y a la vez mejorar la calidad de vida de personas discapacitadas o con problemas físicos.

Matlab es una excelente herramienta que reduce la complejidad del procesado de señales de audio. Su eficiencia computacional permite digitalizar las señales con precisión y tratarlas conforme a las necesidades.

Los coeficientes Cepstrum aportan cuantiosa información sobre un fragmento de audio, con la cual se puede determinar si otra señal se corresponde con el mismo sonido, producido además por la misma persona o no.

El algoritmo K-NN implementado con la medida de distancia euclídea es sencillo de implementar y eficaz a la hora de programar un algoritmo de búsqueda.

El sistema de reconocimiento de hablante que implica todas las técnicas anteriores tiene como resultado un alto porcentaje de acierto si se configura con los parámetros adecuados, concluyendo por tanto que es un buen sistema.

Que sea un buen sistema no implica que sea el mejor. En el mundo de las telecomunicaciones y muy en particular en el mundo de la inteligencia artificial siempre existe la posibilidad de crear un sistema mejor, con mayor precisión y probabilidad de acierto. Este proyecto sólo abarca un pequeño margen de investigación que podría seguir su curso con otras muchas posibilidades, algunas de las cuales se redactan a continuación:

- En la extracción de características sólo se tienen en cuenta los coeficientes cepstrales. Podría estudiarse la combinación con otras características para obtener aún más información. Existen investigaciones que diferencian tonos de voz o sentido del humor, que podría mejorar este tipo de sistemas.

- La distancia euclídea es simple y precisa, pero existen otras distancias como la distancia de Minkowski o la distancia de Mahalanobis, que tienen en cuenta la correlación entre variables aleatorias.

- Existen numerosos algoritmos de clasificación nombrados en el capítulo 2.1, los cuales podrían proporcionar mejores resultados que el K-NN a la hora de relacionar vectores de coeficientes.

- En la toma de decisión podría ponderarse la solución de cada trama en función de su distancia. Es decir, que los vectores con menor distancia elegidos por trama tengan un peso mayor que los de gran distancia a la hora de decidir el hablante final.

- En este trabajo sólo se han introducidos audios de personas con las que el sistema ya estaba entrenado. Podría implementarse un algoritmo que detectara si el audio introducido no coincide con ninguno de los estudiados. Fijar límites superior e inferior en la toma de distancias entre vectores sería una posible solución.

CONCLUSION AND FUTURE WORKS

The speaker recognition system suggests a big advance in technology that allows communication between humans beings and machines. The study of these systems can improve the quality of many devices and at the same time the quality of life for disabled people.

Matlab program is an excellent tool that reduces the complexity of the audio signal processing. It is computationally efficient and is able to digitize signals accurately and treat the according to the needs.

Cepstrum coefficients provide substantial information about an audio clip, with which you can determine if another signal corresponds to the same sound produced by the same person or not.

The K-Means algorithm implemented with the Euclidean distance measure is simple to implement and effective in programming a search algorithm.

The speaker recognition system, involving all the above techniques as a result has a high success rate when configured with the appropriate parameters. We can conclude therefore that it is a good system.

To say it is a good system, doesn´t make it the best. In the telecommunication world and particularly in the world of artificial intelligence there is always the possibility of creating a better system, more accurately and with more probability of success. This project covers only a small region of investigation that could take its course with many other possibilities, some of which are written below:

- The extraction feature only takes into account the Cepstral coefficients. It could combine other features to obtain even more information. There is a research that differentiate tones of voice or sense of humor, which could improve such systems.
- The Euclidean distance is simple and accurate, but there are other distances such as Minkowski distance or Mahalanobis distance, working with the correlation between random variables.

- There are numerous classification algorithms noted in Chapter 2.1 which could provide better results than K-Means when linking coefficient vectors.

- The best decision could be a balanced solution of each frame based on their distance. So the vectors with less chosen distance per frame have greater weight than larger distance when deciding the final speaker weight.

- This paper has only introduced audios of people that the system was already trained to detail. An algorithm could be implemented that will detect if the audio entered does not match any of those studied. Fixing the upper and lower distances between vectors making boundaries would be a possible solution.

BIBLIOGRAFÍA Y REFERENCIAS

Referencias físicas

- **[BEL]** F. Bellesi y F. Ortiz "Reconocimiento de voz para aplicación en domótica" 2008.
- **[CID]** J.Cid "Universidad Carlos III. Teoría de Reconocimiento de Patrones".
- **[DAV]** K.H. Davis, R. Biddulph y S. Balashek "Automatic Recognition of Spoken Digits", Journal of Acoustic Society of America, Vol. 24, 1952.
- **[FUR]** S. Furui "Talker recognition by long time averaged speech spectrum" Electronics and Communications, Japan, 1972.
- **[JUA]** B.H. Juang "The Past, Present, and Future of Speech Processing", IEEE Signal Processing Magazine, mayo 1998.
- **[MAT]** T. Matsui y S. Furui "Concatenated phoneme models for text-variable speaker recognition," Proc. ICASSP ,1996.
- **[PAD]** J. Padrell-Sendra, D. Martın-Iglesias y F. Díaz-de María "Support vector machines for continuous speech recognition" European Signal Processing, Florencia, Italia, 2006.
- **[PAS]** D. Pascual, F. Pla y S. Sánchez, "Algoritmos de agrupamiento" Departamento de Computación, Universidad de Oriente, Cuba.
- **[RAB]** L.R. Rabiner y B.H. Juang "Fundamentals of Speech Recognition" Prentice-Hall, Englewood Cliffs, 1996.

Enlaces virtuales

- **[APP]** Apple, Siri
  www.apple.com/es/ios/siri
- **[BOE]** Boletín Oficial del Estado
  www.boe.es
- **[GOO]** Ok Google
  https://chrome.google.com/webstore/detail/google-voice-search-hotwo/
- **[HTK]** Reconocimiento de voz usando HTK
  http://bibing.us.es/proyectos/abreproy/11529
- **[IBM]** IBM, ViaVoice
  http://www-01.ibm.com/software/pervasive/viavoice.html
- **[INT]** Inteligencia artificial
  http://www.monografias.com/trabajos16/inteligencia-artificial-historia/inteligencia-artificial-historia.shtml
- **[MED]** MedCiencia
  www.medciencia.com
- **[MIC]** Microsoft, SAPI
  http://msdn.microsoft.com/en-us/library/hh361572.aspx
- **[NUA]** Nuance
  www.nuance.es
- **[PER]** Pervoice
  http://www.pervoice.it/Tecnologia
- **[PHI]** Philips
  www.dictation.philips.com/es
- **[RAE]** Real Academia Española
  www.rae.es
- **[REC]** Reconocimiento de Locutor basado en Procesamiento de Voz
  www.fceia.unr.edu.ar/prodivoz/speaker_verification.pdf

- **[REP]** Repositorio Matlab. Melcespst
  www.ee.ic.ac.uk/hp/staff/dmb/voicebox/doc/voicebox/melcepst.html

- **[SAI]** Sail Labs
  http://www.sail-labs.com/products-solutions/commercial-products/media-mining-indexer.html

- **[SEC]** Sociedad de Estudiantes de Ciencia de la Computación.
  http://148.204.64.201/

- **[TUT]** Tutorial básico para el manejo de señales en Matlab
  http://musica.unq.edu.ar/personales/ebonnier/cam2/matlab

- **[WIK]** Wikipedia
  www.wikipedia.com

- **[YOU]** Youtube
  www.youtube.com