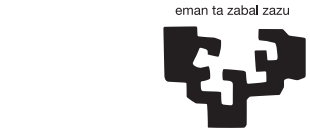
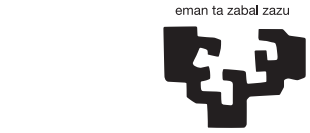


UNIVERSIDAD DEL PAÍS VASCO - EUSKAL HERRIKO UNIBERTSITATEA

TÉCNICAS DE MEJORA DEL RENDIMIENTO DE LOS SISTEMAS DE DIARIZACIÓN DE LOCUTORES

Tesis doctoral presentada por David Tavárez Arriba

Dirigida por Dra. Eva Navas Cordón



UNIVERSIDAD DEL PAÍS VASCO - EUSKAL HERRIKO UNIBERTSITATEA

TÉCNICAS DE MEJORA DEL RENDIMIENTO DE LOS SISTEMAS DE DIARIZACIÓN DE LOCUTORES

Tesis doctoral presentada por David Tavárez Arriba

Dirigida por Dra. Eva Navas Cordón

El doctorando El director

Bilbao, diciembre 2016

Técnicas de mejora del rendimiento de los sistemas de diarización de locutores

Author: David Tavárez Arriba
Advisor: Dra. Eva Navas Cordón

Text printed in Bilbao
First edition, diciembre 2016

A todos los que me apoyaron durante la realización de esta tesis.

Abstract

The main objective of the speaker diarization is the division of an audio signal in the different speakers who appear in it. Speaker diarization therefore seeks to answer the question of "who spoke when," by identifying the start and end points of the interventions of these speakers. This diarization task is typically used as a first processing step in various speech technologies, such as automatic speech recognition, speaker verification or audio indexing.

To accomplish this task, it is required to turn to several algorithms with quite different purposes that often run sequentially, such as parameterization, audio segmentation, speaker segmentation, speaker clustering or resegmentation.

The work developed in this thesis tries to cover various aspects related to the different algorithms involved in the diarization process. Several new techniques have been analyzed independently and the most commonly used methods in the corresponding areas of study have been used to compare the results obtained. Besides, databases, architectures and appropriate metrics have also been used for each particular area.

A first important aspect of this study has been the gathering of databases that enable the development of quality diarization systems. The multilingual databases found in the literature generally have different speakers for the different languages or have been designed for other speech processing areas. This thesis tries to cover this gap by creating two new databases: Ahonews in the audio broadcast domain and Ahomeetings in recorded meetings domain.

Audio segmentation analysis has mainly focused on the broadcast domain and the segments with low-level background music problem. To

deal with this problem, a new technique that is based on the postprocessing of the speech segments has been developed. Furthermore, a second technique that uses i-vectors to perform a refinement of the speech segments in challenging conditions has been introduced. The results obtained in both cases have been successfully validated in different audio segmentation evaluation campaigns organized by the "Red Temática en Tecnologías del Habla".

The study has then focused on the analysis of the different label level fusion methods and their poor behavior in conditions of class imbalance in the database. A new label fusion method which takes into account the mentioned imbalance between the classes has been developed as a solution to this problem. The proposed method shows a significant improvement of the results in different areas of the speech processing, such as audio segmentation, emotion recognition or speaker recognition and verification.

In regard to the speaker segmentation task, this thesis introduces a speaker turn detection technique based on frame by frame analysis that reduces the system delay. Thus, the intention is to extend the online performance to the traditional diarization systems, which generally need the complete signals before the processing starts. The experiments performed on the meeting domain present better detection results for the developed method than for the traditional BIC.

A final analysis presents the resegmentation as an alternative to solve some particular problems related to the diarization systems. In this aspect, a new technique for diarization improvement based on the identification and reunification of the clusters belonging to the same speaker has been introduced. The results obtained in different experiments have shown the good performance of the developed technique.

Resumen

La diarización de locutores (*"who spoke when"*, quién habló cuándo) tiene como objetivo principal la división de una señal de voz en los diferentes locutores que aparecen en ella, identificando los puntos de inicio y final de cada una de las intervenciones de dichos locutores. Habitualmente se utiliza como una primera etapa de procesado en distintas tecnologías de la voz, como reconocimiento automático del habla, verificación de locutor o indexado de audio.

Para llevar a cabo esta tarea, resulta necesario recurrir a distintos algoritmos con diferentes finalidades que a menudo se ejecutan de forma secuencial, como son parametrización, segmentación de audio, segmentación de locutor, agrupación de locutores o resegmentación.

Mediante el trabajo desarrollado en esta tesis se pretende cubrir diferentes aspectos relacionados con los distintos algoritmos involucrados en el proceso. Se presenta en cada caso un análisis de las técnicas más comúnmente utilizadas y de los nuevos métodos desarrollados, comparando los resultados obtenidos mediante bases de datos, arquitecturas y métricas adecuadas para cada área de estudio en particular.

Un aspecto importante ha sido por tanto, la recopilación de bases de datos que permitan el desarrollo y la implementación de sistemas de diarización en los distintos ámbitos de aplicación. Además, las bases de datos multilingües recogidas en la literatura generalmente presentan diferentes locutores para los distintos idiomas o han sido diseñadas para realizar tareas en otras áreas del procesado de voz, por lo que se ha tratado de llenar este vacío mediante la creación de dos nuevas bases de datos: Ahonews en el entorno de difusión de audio y Ahomeetings en el entorno de reuniones de trabajo.

El análisis de la segmentación de audio se ha centrado en el entorno de difusión y la problemática de los segmentos con música de fondo de bajo nivel. Para lidiar con este problema se ha diseñado una nueva técnica basada en el postprocesado de segmentos de voz. Además, se ha presentado una segunda técnica basada en i-vectors que permite el refinamiento de los segmentos de voz en entornos complejos. Las conclusiones resultantes en ambos casos han sido validadas con éxito en distintas campañas de evaluación de segmentación de audio organizadas por la Red Temática en Tecnologías del Habla.

Una fase posterior del estudio ha centrado el análisis en los distintos métodos de fusión de etiquetas, prestando especial interés al pobre comportamiento general obtenido en condiciones de desequilibrio en la base de datos. Como solución a esta problemática, se ha desarrollado un método de fusión de etiquetas que tiene en cuenta el mencionado desequilibrio entre clases. Los resultados obtenidos muestran un mejor comportamiento del método propuesto en distintas áreas del procesado de la voz, como son la segmentación de audio, el reconocimiento de emociones o el reconocimiento y verificación de locutores.

En cuanto a la tarea de segmentación de locutores, se ha introducido en esta tesis una técnica de detección de cambios de turno basada en el análisis trama a trama, que permite reducir el retardo introducido por el sistema. De esta forma, se pretende extender el funcionamiento online a los sistemas de diarización, que tradicionalmente precisan de las señales completas al empezar el procesado. Los experimentos realizados en entorno de reuniones muestran una mejor detección por parte del método desarrollado frente a un método clásico como es el BIC.

Un último análisis evalúa la resegmentación como alternativa a la hora de solucionar problemas concretos de los sistemas de diarización. En este aspecto, se ha propuesto una técnica de mejora de la diarización basada en la identificación y reagrupamiento de clusters pertenecientes a un mismo locutor. Los resultados obtenidos en diversos experimentos han demostrado el buen funcionamiento de la técnica desarrollada.

Laburpena

Esatari diarizazioak (“who spoke when”, zeinek hitz egin zuen noiz) ahots seinale batean agertzen diren esatari desberdinen banaketa du helburu nagusitzat, esatari bakoitzaren parte-hartzeen hasiera eta bukaera identifikatuz. Normalean hizketa prozesamenduaren lehen etapan erabiltzen da, esaterako, hizketa-ezagutze automatikoan, esatari egiaztatze edo audio indexazioan.

Zeregin hau burutzeko, sekuentzialki exekutatzen diren helburu anitzeko algoritmoez baliatu behar da, hala nola, parametrizazioa, segmentazioa, esatari segmentazioa, esatari taldekatzea edo birsegmentazioa.

Tesi honetan egindako lanarekin, prozesu honetan erabiltzen diren algoritmo desberdinekin erlazionaturiko alderdien berri emango da. Kasu bakoitzerako, normalean erabiltzen diren eta garatutako metodoen analisiak aurkeztuko dira, ikerketa eremu bakoitzerako egokiak diren datu-base, arkitektura eta metrika bidez lortutako emaitzak konparatuz.

Hori dela eta, diarizazio sistemak aplikazio esparru desberdinetan inplementatu eta garatzea baimentzen diguten datu-basearen bilketa garrantzitsua izan da. Gainera, literaturan jasotako datu-base eleaniztunek esatari desberdinak erabiltzen dituzte hizkuntza desberdinetarako edo hizketaren prozesamenduko beste eremu batzuetarako daude prestaturik. Hutsune hau betetzeko bi datu-base berri sortu dira: Ahonews audioaren difusio-inguruan eta Ahomeetings laneko bilera-inguruan.

Audioaren segmentazio analisia difusio-inguruan eta atzean maila baxuko musika duten segmentuetan zentratu da. Arazo honi aurre egiteko ahots segmentuen postprozesamenduan oinarritutako teknika bat diseinatu da. Gainera, i-vector-etan oinarritutako bigarren teknika bat

aurkeztu da inguru konplexuetan ahots segmentuen fintze bat ahalbidetzen duena. Bi kasuetan lortutako ondorioak era arrakastatsuan balioztatu dira Red Temática en Tecnologías del Habla-k antolatutako audio segmentazio ebaluazioetan.

Ikerketaren ondorengo faseak etiketen fusioan ardaztu du analisia, datubaseen desoreka egoeran lortutako portaera txarrean interes berezia jarriz. Problematika honen soluzio bezala, klaseen arteko desoreka kontuan hartzen duen etiketa fusio metodo bat garatu da. Lortutako emaitzek proposatutako metodoaren portaera hobea erakusten dute hizketaren prozesamenduaren eremu ezberdinetan, hala nola, audio segmentazioa, emozio antzematea edo esatari ezagutze edo antzematea.

Esatari segmentazioari dagokionez, tesi honetan tramatik tramarako analisian oinarritzen den txanda aldaketa detekzio teknika bat aurkeztu da, sistemak sartutako atzerapena txikitzen duena. Era honetan, diarizazio sistemen on-line funtzionamendua zabaldu nahi da, tradizionalki seinale osoak behar dituztenak prozesamendua hasteko. Bilera-inguruan egindako esperimentuek detekzio hobea ematen dute garatutako metodoarekin klasiko den BIC-ek baino.

Azken analisiak birsegmentazioa ebaluatzen du diarizazio sistemen arazo konkretuak konpontzeko tresna bezala erabiliz gero. Alde horretatik, esatari berarenak diren cluster-en identifikazioan eta birbilketan oinarritutako diarizazioa hobetzeko teknika bat proposatu da. Esperimentu ezberdinek emandako emaitzek erakutsi dute garatutako teknikaren funtzionamendua ona dela.

Acknowledgements

Llegados a este punto quisiera agradecer el esfuerzo y dedicación de todos aquellos que han hecho posible esta tesis.

En primer lugar, quiero expresar mi más sincero agradecimiento a mi directora de tesis, Eva Navas, por el apoyo brindado a lo largo de estos años de doctorado. Has sido directora y compañera a partes iguales, con una labor atenta y comprometida en ambas facetas. Gracias por compartir los buenos momentos y la paciencia en los no tan buenos, porque malo no recuerdo ninguno.

Del mismo modo, quiero dar las gracias a Doña Inmaculada, "The headmaster", por la confianza depositada en mí desde el primer momento. Gracias por ofrecerme formar parte de este grupo y de este proyecto. Sin lugar a duda eres la principal culpable de que nos encontremos hoy aquí.

Asimismo, me gustaría agradecer el apoyo recibido durante estos años por parte de todos mis compañeros de Aholab. A Ibon, por esos momentos solucionando los problemas del metro. A Igor, por estar ahí en mis primeras andaduras en congresos en tierras lejanas. A Jon, por enseñarme lo que significa ser profesor en la escuela. Y a Iñaki, ¿qué sería del laboratorio sin tu paso por Aholab? gracias por tu inestimable ayuda. También quiero agradecer a Iker su paciencia en mis inicios y la impecable documentación de todos sus programas.

Mención aparte merece la ayuda y el apoyo recibido de mis queridos compañeros Umpa Lumpa. Gracias Agustín por cargar conmigo todos estos años, en ocasiones literalmente, sin desesperar tantas veces como posiblemente merecía. Gracias Luis por poner voz a la cordura en los

momentos de oscuridad, siempre serás el líder que marca nuestro camino. No me olvido de ti Daniel. Sabemos tienes el rango de maestro, pero para nosotros, o por lo menos para mí, siempre serás uno más de los umpa lumpa. Gracias por todo tu tiempo. Y gracias también a Xabi, por renovar el espíritu friki del laboratorio, y que tanta falta nos hacía.

Por último, y no por ello menos importante, quiero expresar mi más profundo agradecimiento a mi familia, en especial a mis padres y a mi hermana, por todo el apoyo recibido no sólo en esta etapa de la tesis, sino en cada aspecto importante en mi vida hasta llegar aquí. Gracias por estar siempre ahí. Y por supuesto, a Irantzu. Casualidad o no nuestros caminos se cruzaron al iniciar el doctorado y sé que seguirán unidos mucho después de concluirlo. Gracias por tu apoyo todos estos años.

Eskerrik asko,

David Tavárez.

diciembre 2016.

Índice general

Índice de figuras

Índice de Tablas

El procesado de señal de voz se divide en dos grandes bloques: síntesis y diarización.

David Tavárez

CAPÍTULO

1

Introducción

En los últimos años, el aumento de la velocidad de procesamiento, la capacidad de almacenamiento y el ancho de banda disponible han facilitado la acumulación de grandes volúmenes de audio de distinta naturaleza (emisiones de radio y TV, correos de voz, reuniones de trabajo...). Por ello, existe una creciente necesidad de aplicar diferentes tecnologías del habla que permitan una búsqueda efectiva, indexación y acceso a la información de las diversas fuentes presentes en el audio.

Un archivo de audio es una grabación de uno o varios canales que consta de múltiples fuentes de audio (segmentos de diferentes locutores, segmentos de música, segmentos de ruido, etc.). En términos generales, el objetivo de la diarización es detectar los cambios de locutor en una grabación e identificar qué segmentos de voz corresponden a un mismo locutor, respondiendo de esta forma a la pregunta *"who spoke when"*, quién habló cuándo [144].

La finalidad de un sistema de diarización de locutores es, por tanto, la separación de una señal de voz en los diferentes locutores que aparecen en ella, definiendo sobre la señal disponible los puntos de inicio y final de las intervenciones de cada uno de ellos. Un ejemplo del resultado de este proceso puede verse en la figura 1.1.

En un sentido más amplio, se define diarización como la tarea de marcar y clasificar cada una de las diferentes fuentes que componen un audio de entrada, siendo los detalles propios de cada aplicación específica.

1. INTRODUCCIÓN

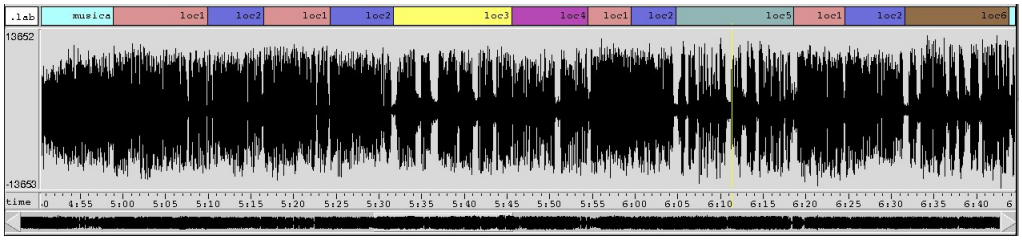


Figura 1.1: Resultado del proceso de diarización de una señal de audio

Para llevar a cabo este cometido es necesario recurrir a múltiples tecnologías del habla [43], como pueden ser: detección de locución (¿Hay alguien hablando?), reconocimiento de género (¿Quien habla es hombre o mujer?), reconocimiento de lengua (¿En qué idioma se está hablando?), reconocimiento del habla (¿Qué se está diciendo?), reconocimiento de locutor (¿Quién está hablando?), etc.

Generalmente, se utiliza la diarización de locutor como paso previo a la aplicación de distintas tecnologías de procesado de la voz, como reconocimiento automático del habla, verificación de locutor o indexado de audio.

Tras una primera división del audio en diferentes clases acústicas (voz, música, ruido...), la segmentación de locutores proporciona segmentos homogéneos a los sistemas de reconocimiento automático de habla, lo que favorece la correcta transcripción del audio procesado. La agrupación de los segmentos de los distintos locutores permite además mejorar el rendimiento de los sistemas de reconocimiento que hacen uso de técnicas de adaptación de locutor.

Del mismo modo, los sistemas de reconocimiento y verificación de locutor precisan de segmentos de voz en los que un único locutor esté presente para proporcionar una salida adecuada. La diarización de locutor se encarga en este caso de llevar a cabo la división del audio en segmentos individuales pertenecientes a cada uno de los locutores presentes en la grabación.

Por otra parte, el etiquetado o “transcripción enriquecida” proporcionado por los sistemas de diarización de audio facilitan el indexado y la localización automática de recursos multimedia pertenecientes a un locutor o locutores determinados.

Respecto a las áreas principales de aplicación de la diarización, se definen tres ámbitos diferentes de trabajo que tradicionalmente han obtenido mayor atención por parte de la comunidad científica:

- Programas de radio y televisión que contienen pausas publicitarias, música, entrevistas, voz con diferentes tipos y niveles de ruido de fondo etc. sobre un único canal. La aplicación del sistema de diarización permite, además de la discriminación de los segmentos de voz frente a segmentos correspondientes a otros eventos, la identificación de los locutores participantes. Estos sistemas suelen incluir adicionalmente la transcripción habla-texto de las señales para la clasificación e indexación automática de las señales.

- Reuniones de trabajo donde varios locutores interactúan en la misma habitación. El sistema de diarización permite el seguimiento del discurso de cada participante y opcionalmente la transcripción del discurso para su posterior indexación y almacenamiento. Generalmente las grabaciones se realizan con varios micrófonos.

- Audio telefónico o conferencias donde varias personas establecen una comunicación a través de una conversación telefónica. En este ámbito las grabaciones involucran generalmente a dos locutores. Además de la segmentación, el sistema de diarización incluye habitualmente tareas de identificación de los locutores presentes en las grabaciones.

Cada una de estas áreas de aplicación presenta una problemática y unos retos diferentes. Sin embargo hay cuestiones comunes que aún no han sido resueltas y que actualmente centran gran parte de los esfuerzos de investigación en este campo, como son la segmentación de audio en entornos complejos o el funcionamiento online de los sistemas de diarización.

El trabajo realizado en esta tesis abarca diferentes aspectos relacionados con los distintos algoritmos involucrados en el proceso de diarización de locutores, centrando el esfuerzo en los problemas comunes a los diferentes campos de aplicación, analizando en cada caso las distintas técnicas recogidas en la literatura y presentando nuevos métodos desarrollados para lidiar con dicha problemática.

1.1 Descripción de un sistema de diarización básico

El objetivo principal de un sistema de diarización de locutores es detectar los cambios de locutor en una grabación e identificar los segmentos de voz que corresponden a un mismo locutor, respondiendo a la pregunta quién habla cuándo [144]. Adicionalmente, si se dispone de la información necesaria es posible llegar a identificar cada uno de ellos. Para ello, se combinan habitualmente varios algoritmos con diferentes finalidades que suelen ejecutarse de forma secuencial [43].

En la Figura 1.2 se muestra un diagrama con los bloques más habituales en un sistema de diarización: parametrización, detección de actividad vocal, detección de cambio de turno y agrupación de locutores.

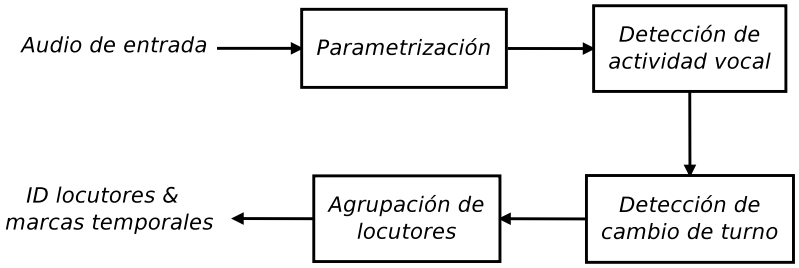


Figura 1.2: Esquema básico de un sistema de diarización de locutores

Se denomina **parametrización** al proceso de extracción de una serie de características que representan la señal de voz, es decir, contienen la información más relevante de la señal. Estas características pueden ser de diferente naturaleza (energía, frecuencia, contenido armónico...), y dado que la voz no es una señal estacionaria, sus características varían en intervalos de corta duración, por lo que nuevos parámetros deben ser extraídos cada cierto tiempo.

Para ello, en primer lugar, se enventana un segmento de la señal y se parametriza. A continuación se desplaza la ventana repitiendo el proceso para cada uno de los segmentos [156]. Las técnicas de parametrización más ampliamente utilizadas en el campo de la diarización de locutores son los Mel Frequency Cepstral Coefficients (MFCC), Linear Prediction Cepstral Coefficients (LPCC) y los Perceptual Linear Prediction Cepstral Coefficients (PLPC) [4]. Todas ellas están basadas en información espectral obtenida a partir de segmentos de voz de corta duración (short-term).

El bloque de **detección de actividad vocal** se encarga de descartar los segmentos de audio que no contienen voz, de forma que las siguientes etapas produzcan menos errores. Dependiendo del entorno de trabajo del sistema de diarización, la señal de audio correspondiente a los segmentos de "no voz" puede proceder de gran variedad de eventos acústicos: silencio, ruido, música, aplausos, etc., por lo que a menudo la detección de actividad vocal deriva en un análisis más complejo del audio denominado segmentación de audio.

El enfoque más habitual en esta etapa consiste en realizar una segmentación mediante el algoritmo de Viterbi con modelos de mezclas de gaussianas (GMM) entrenados con datos previamente etiquetados. En ocasiones se utilizan también modelos más elaborados como modelos ocultos de Markov (HMM) multiestado, máquinas de vectores de soporte o redes neuronales. En principio, es posible utilizar sólo dos modelos (voz y no voz), sin embargo, resulta más conveniente contar con modelos específicos en el caso de que diferentes eventos acústicos estén presentes en el audio. Habitualmente se utilizan modelos para ruido, música, voz limpia, voz con ruido y voz con música [116]. También es posible entrenar diferentes modelos para voz femenina y masculina o para diferenciar voz de banda ancha y estrecha si se dispone de suficiente material.

La **detección de cambio de turno** es un paso crítico en un sistema de diarización de locutores. Una vez descartados los segmentos que no contienen voz, se identifican en esta etapa las fronteras entre locutores, es decir, los puntos donde se produce un cambio de locutor. Habitualmente esta etapa recibe el nombre de segmentación de locutores, ya que su objetivo principal es dividir el audio en segmentos correspondientes a los distintos locutores presentes en la grabación.

La mayor parte de sistemas de diarización realizan esta tarea por medio de alguna distancia entre dos ventanas adyacentes de señal de audio. Si esta distancia supera un determinado umbral, se asume que existe un cambio de locutor entre las dos ventanas. La diferencia entre los distintos algoritmos reside principalmente en la métrica utilizada para el cálculo de la distancia y el esquema de enventanado. Una de las métricas más utilizadas en detección de cambio de locutor es el criterio de información Bayesiana o BIC [26]. La razón de probabilidad generalizada (Generalized Likelihood Ratio,GLR) [93], la divergencia gaussiana [133] o la T^2 de Hotlling [161] son ejemplos de otras medidas utilizadas. Generalmente

pueden resultar menos precisas, pero requieren menor tiempo de cálculo. Un enfoque alternativo consiste en utilizar métricas de menor coste computacional para posteriormente refinar los resultados obtenidos mediante BIC [34].

El objetivo del bloque de **agrupación de locutores** (o *clustering* en inglés) es determinar qué fragmentos de voz pertenecen al mismo locutor. Tras la división del audio en segmentos homogéneos consecuencia de la detección de cambio de turno, en esta etapa se agrupan los segmentos que presentan características acústicas similares, susceptibles de pertenecer a un mismo locutor. Habitualmente esta etapa define el número de locutores diferentes encontrados por el sistema en el audio.

Se implementa generalmente como un proceso de agrupación de abajo arriba. En primer lugar, se calcula una medida de distancia entre cada par de clusters y se combina el par que presenta menor valor. A continuación, se actualiza la matriz de distancias y se selecciona un nuevo par hasta que se cumplen las condiciones de un determinado criterio de parada. De nuevo en este caso pueden utilizarse distintas medidas de distancia, como son BIC, GLR, la distancia entre GMMs, etc.

Por último, es habitual encontrar en estos sistemas una etapa de **resegmentación** o postprocesado de las marcas [8] [13]. En este punto, se dispone posiblemente de suficiente información de los locutores presentes en el audio para entrenar modelos robustos para cada uno de ellos y utilizarlos en combinación con los modelos de "no voz" para realizar una nueva segmentación y refinar las fronteras establecidas por el sistema de diarización. Este proceso se puede repetir iterativamente para mejorar los resultados.

1.2 Motivación

El objetivo global que se plantea en esta tesis es el desarrollo de nuevas técnicas que mejoren el rendimiento de los sistemas de diarización de locutores, centrando el esfuerzo en problemas comunes a los diferentes campos de aplicación de este tipo de sistemas, como son, una mejor clasificación del audio en entornos complejos y el funcionamiento online de dichos sistemas.

El funcionamiento online de los sistemas de diarización pretende proporcionar información visual a posibles oyentes discapacitados sobre quién es la persona que está hablando en cada momento (con objeto de que pueda leerle los labios), por lo que el estudio en este caso se centrará principalmente en el dominio de las reuniones de trabajo, priorizando este dominio frente a otros tales como los sistemas de radiodifusión o las conversaciones telefónicas.

El hecho de tener que tomar decisiones sobre la identidad de los hablantes sobre la marcha, introduce un grado de dificultad adicional sobre los sistemas clásicos utilizados por lo general para la indexación y clasificación de material de audio. Por otro lado, las reuniones de trabajo se producen frecuentemente entre habituales, lo que introduce un nuevo factor, el conocimiento previo de todos o parte de los participantes, que deberá ser aprovechado por las distintas técnicas que se desarrollen, basadas en clasificación de los locutores o en la fusión de la salida de varios sistemas de reconocimiento independientes.

La mejora de clasificación del audio centrará el análisis en el entorno de difusión de noticias, donde segmentos de música o ruido aparecen frecuentemente junto a los segmentos de voz. El objetivo principal en este caso será el diseño y la implementación de distintas técnicas que permitan la extracción de dichos segmentos de voz. Estos segmentos serán entregados posteriormente a las distintas etapas de los sistemas de diarización, por lo que una mejor clasificación del audio supondrá la mejora en el rendimiento de dichos sistemas.

Un aspecto importante para el desarrollo del sistema de diarización es la base de datos a utilizar. Por ello, es también objetivo de este trabajo investigar sobre las bases de datos disponibles para esta tarea, y realizar si fuera necesario bases de datos propias adecuadas para esta tesis en particular. Sin embargo, no debe olvidarse que los resultados a obtener tienen que ser contrastables, por lo que debe ser

objetivo del trabajo desarrollado la asistencia a las campañas de evaluación que en este campo se desarrollan organizadas por el NIST y Albayzin. Para ello será necesario realizar los ajustes necesarios en los distintos sistemas implementados para que sean también aplicables al tipo de aplicación que plantee la evaluación.

Así pues, los objetivos propuestos a desarrollar en esta tesis pueden resumirse finalmente en los siguientes:

- Obtención y/o elaboración de una base de datos adecuada para el desarrollo del trabajo a realizar.
- Desarrollo de algoritmos y técnicas que permitan el desarrollo en tiempo real del proceso de diarización.
- Desarrollo de algoritmos innovadores que aprovechen el conocimiento previo de parte de los participantes en la reunión.
- Desarrollo de técnicas de mejora de la clasificación del audio en el entorno de difusión de noticias que faciliten la extracción de los segmentos de voz.
- Evaluación contrastable de los sistemas desarrollados, adaptándolos para participar en las campañas de evaluación organizadas por el NIST y Albayzin.

1.3 Esquema de la tesis

El capítulo 2 recoge las bases de datos utilizadas durante el desarrollo de esta tesis. En primer lugar se describen las bases de datos encontradas en la literatura, proporcionadas habitualmente con motivo de la participación en distintas campañas de evaluación o challenges, y que han sido utilizadas como marco de los múltiples experimentos llevados a cabo en esta tesis. En segundo lugar, se presentan las dos bases de datos creadas durante el desarrollo del trabajo realizado, y que forman parte en sí mismas de las aportaciones de esta tesis.

Los capítulos 3-6 conforman el bloque principal de la tesis. En ellos se describen las aportaciones realizadas en las distintas etapas implicadas en el proceso de diarización de locutores. Cada uno de estos capítulos presenta el estado del arte referente a las distintas áreas involucradas, las distintas técnicas de mejora propuestas y los experimentos realizados para validar dichas técnicas, así como una sección final donde se recogen las principales conclusiones extraídas en cada caso.

El capítulo 3 describe los diferentes procedimientos desarrollados en el ámbito de la segmentación de audio. En primer lugar se presentan las técnicas utilizadas habitualmente en la literatura, así como los métodos aplicados en la evaluación de los sistemas de segmentación. A continuación, se describen las técnicas de mejora de la segmentación propuestas y los resultados obtenidos en las distintas campañas de evaluación utilizadas como marco de validación de dichas técnicas.

El capítulo 4 se centra en el estudio de la fusión de clasificadores, y más concretamente, de la fusión a nivel de etiqueta. En la primera parte del capítulo se analizan los métodos de fusión de etiquetas más ampliamente utilizados, así como el reto que plantean las bases de datos desequilibradas a dichos métodos. A continuación, se describe el algoritmo de fusión propuesto para lidiar con el problema del desequilibrio en la base de datos utilizada y los experimentos llevados a cabo en la validación del mismo. En este capítulo se incluyen además distintos experimentos en diferentes áreas de aplicación, como son la segmentación de audio, reconocimiento de locutores o reconocimiento de emociones a partir de la voz.

El capítulo 5 recoge los esfuerzos realizados en el ámbito de la segmentación de locutores. Al igual que en los capítulos anteriores, se realiza en primer lugar un análisis de los distintos métodos de segmentación de locutores presentes en

la literatura y las métricas utilizadas para llevar a cabo la evaluación de los distintos métodos. A continuación se describe la técnica de segmentación propuesta, orientada a la detección online de cambios de locutor. Por último, se muestran los resultados obtenidos al aplicar dicha técnica a diferentes bases de datos utilizadas habitualmente en el área de la diarización de locutores.

El capítulo 6 presenta distintos enfoques destinados a la mejora de las marcas proporcionadas por un sistemas de diarización. Primero se revisa la literatura referente a los diferentes métodos de análisis y postprocesado orientados a la mejora de distintos sistemas de diarización. Posteriormente, se describe la técnica de postprocesado de marcas propuesta para tratar distintos problemas presentes en determinados sistemas de diarización de locutores. Por último, se muestran los resultados obtenidos mediante la aplicación de la técnica propuesta a diversos sistemas de diarización y distintas bases de datos.

Por último, el capítulo 7 recoge las conclusiones generales derivadas de este trabajo y los posibles trabajos futuros a llevar cabo. En primer lugar se resumen las distintas aportaciones de la tesis en las distintas áreas de trabajo involucradas. Para finalizar, un último punto recoge la difusión de los resultados obtenidos durante el desarrollo de la tesis realizada.

Los datos no son información, la información no es conocimiento, el conocimiento no es comprensión, la comprensión no es sabiduría.

Clifford Stoll

CAPÍTULO

2

Bases de datos

Del mismo modo que ocurre en cualquiera de las diferentes tecnologías de voz, hay una necesidad de bases datos transcritas que permitan el desarrollo y la implementación de sistemas de diarización de locutores de calidad.

La evaluación de los distintos algoritmos desarrollados mediante bases de datos relevantes es un aspecto esencial en la evaluación del progreso y en el descubrimiento de nuevas dificultades a resolver en los sistemas aún por desarrollar [38]. El establecimiento de un marco de evaluación común supone además un punto clave a la hora de comparar los sistemas desarrollados por los distintos laboratorios.

Este capítulo recoge las bases de datos utilizadas a lo largo del trabajo realizado en esta tesis. En primer lugar se presentan distintas bases de datos de terceros, recopiladas generalmente mediante la participación en distintas campañas de evaluación (Albayzin, Aibo Emotion Corpus, NIST i-vector challenge...), así como diferentes bases de datos ampliamente utilizadas en los distintos campos de aplicación de la tesis (UCI Machine Learning Repository, ICSI Meeting Corpus, AMI Meeting Corpus...). A continuación se introducen dos nuevas bases de datos en español y euskera, diseñadas para el desarrollo de tareas de diarización de locutor en dos entornos de aplicación diferentes: Ahonews en el ámbito de la difusión de noticias y Ahomeetings en el ámbito de las reuniones de trabajo.

2.1 Bases de datos de terceros

2.1.1 Albayzin 2010 speaker diarization evaluation

La base de datos proporcionada en la campaña de evaluación Albayzin 2010 [158], grabada por el grupo de investigación TALP de la UPC y etiquetada por Verbio Technologies, contiene grabaciones de tipo broadcast pertenecientes al canal de televisión catalán 3/24 que incluyen reportajes, anuncios, entrevistas, debates, así como retransmisiones en directo a pie de la noticia.

A partir de los vídeos originales se extrajeron las grabaciones de audio con una tasa binaria inicial de 32 kHz y una resolución de 16 bits. Posteriormente, un proceso de diezmado determinó la tasa binaria definitiva de 16 kHz para dichas grabaciones. Un total de 24 archivos de audio, con una duración aproximada de 88 horas, fueron proporcionados por la organización de la evaluación, 8 de los cuales, con aproximadamente de 30 horas de duración, fueron seleccionados para conformar la parte test de la base de datos.

El catalán es el idioma principal en esta base de datos, aunque se estima una proporción del 8.5 % de segmentos de voz en castellano. El número de locutores que intervienen en cada una de las grabaciones varía desde 30 hasta 250. La tabla 2.1 muestra la distribución de dichos locutores en la base de datos. Se puede observar un desequilibrio considerable en la participación de hombres y mujeres en las grabaciones, tanto en el número de locutores, como en el número de segmentos y la duración total de los mismos. Aproximadamente el 37 % de la base de datos corresponde a voz limpia, 5 % a música, 15 % a voz con música de fondo, 40 % a

Tabla 2.1: Distribución de locutores en la base de datos Albayzin 2010

Género	Locutores	Duración (h.)	Segmentos
Hombres	1239	44:23:41	12869
Mujeres	507	25:43:54	7559
Desconocido	270	07:50:38	2579
Solapado	68	00:12:38	241

Tabla 2.2: Distribución de los segmentos de voz en función del canal y las condiciones de fondo en la base de datos Albayzin 2010

Canal	Voz limpia	Voz de fondo	Música de fondo	Ruido de fondo
Desconocido	04:27:10	00:18:54	04:36:06	01:15:30
Estudio	15:04:24	01:36:16	08:40:47	00:57:12
Teléfono	00:00:40	00:00:10	-	00:06:47
Exteriores	14:49:44	03:55:29	01:52:52	18:55:19

voz con ruido de fondo, además de un 3 % de "otros", donde se engloba todo el material que no pertenece a las cuatro clases anteriores, incluyendo el ruido. La duración total de los segmentos de voz en función de las condiciones específicas de fondo se recogen en la Tabla 2.2.

2.1.2 Albayzin 2012 audio segmentation evaluation

Los organizadores de la campaña de 2012 [102] proporcionaron en este caso dos bases de datos de audio diferentes, pertenecientes nuevamente al entorno de programas de noticias, para ser utilizadas en el desarrollo de los sistemas.

La primera base de datos, utilizada en la campaña Albayzin 2010, está formada por las grabaciones de programas emitidos por el canal catalán de televisión 3/24.

La segunda base de datos proporcionada proviene de la Corporación Aragonesa de Radio y Televisión (CARTV), que donó parte de su archivo de Aragón Radio para fines educativos y de investigación.

Está formada por unas 20 horas de audio con la distribución de clases que se describe a continuación: 22 % de voz limpia, 9 % de música, 31 % de voz con música de fondo, 26 % de voz con ruido de fondo y 12 % de otros, entendiendo la clase "otros" como silencios, ruido y combinaciones de clases no mencionadas. Aproximadamente 4 horas se destinaron al desarrollo de los distintos sistemas, mientras que las 16 horas restantes fueron definidas por la organización como test.

Todas las grabaciones se proporcionan en formato PCM, mono, con 16 bits de resolución y 16 kHz de frecuencia de muestreo.

2.1.3 Albayzin 2014 audio segmentation evaluation

La base de datos proporcionada por la organización de la campaña de evaluación Albayzin 2014 [22] surge de las combinación de tres bases de datos diferentes:

- La base de datos de noticias en catalán del canal de televisión 3/24 utilizada en las evaluaciones de segmentación de audio de 2010 y 2012.
- La base de datos de Aragón Radio perteneciente a la Corporación Aragonesa de Radio y Televisión utilizada en la evaluación de 2012.
- Sonidos ambientales obtenidos de Freesound.org [46] y HuCorpus [61] para ser fusionados con las grabaciones de los canales 3/24 y Aragón Radio.

Como resultado de la combinación, se proporcionan 35 nuevas grabaciones. Las 20 primeras grabaciones, con una duración aproximada de 21 horas fueron destinadas al entrenamiento de los sistemas. Las 15 restantes, con aproximadamente 15 horas de audio disponible, fueron designadas por la organización para formar la parte de test de la base de datos.

En la Figura 2.1 se muestra la distribución de las distintas clases de audio en la parte de entrenamiento de la base de datos. Se puede observar cómo las clases que contienen voz representan más de 92 % del audio disponible. Por el contrario, se observan dos clases minoritarias, ruido aislado y música con ruido, con una presencia inferior al 0.3 % y 0.5 % respectivamente.

El formato elegido nuevamente para las grabaciones proporcionadas es PCM, mono, little endian con 16 bits de resolución y 16 kHz de frecuencia de muestreo.

2.1.4 UCI Machine Learning Repository

Se trata en realidad de un repositorio público que contiene una amplia colección de bases de datos de distinta naturaleza, tanto reales como sintéticos [83].

Creado inicialmente como un directorio FTP por David Aha y diversos estudiantes de la Universidad de California, Irvine, ha visto aumentada su acogida por la comunidad científica en los últimos años a la hora de llevar a cabo experimentos en el campo del aprendizaje automático.

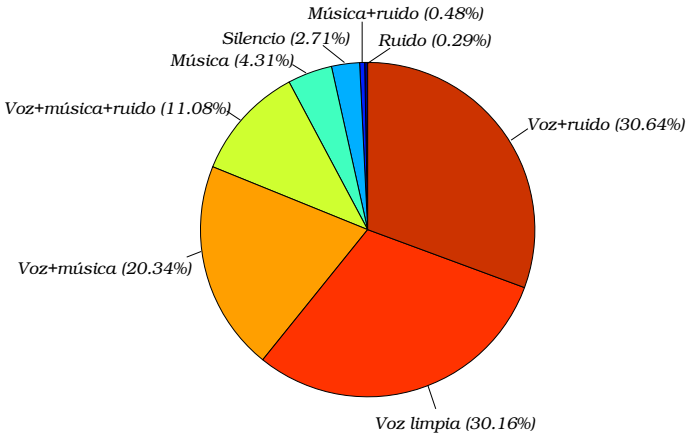


Figura 2.1: Distribución del audio en la base de datos Albayzin 2014

Citado en más de 1000 ocasiones, es uno de los 100 artículos más populares en el ámbito del las ciencias informáticas. Grupos de investigación de todo el mundo utilizan el repositorio como fuente primaria de bases de datos que permitan evaluar sus algoritmos, lo que facilita la comparación de los distintos métodos desarrollados en el campo del aprendizaje automático.

La versión actual de la pagina web del repositorio, diseñada por Arthur Asuncion y David Newman en 2007, cuenta con aproximadamente 350 bases de datos diferentes, disponibles a través de una sencilla interfaz de búsqueda.

2.1.5 FAU Aibo Emotion Corpus

La base de datos AIBO [136] contiene cerca de 9 horas de grabaciones realizadas a 51 niños de entre 10 y 13 años, mientras jugaban con el robot de Sony AIBO. Dichas grabaciones fueron realizadas en dos colegios diferentes, Ohm (26 niños) y Mont (25 niños), por lo que la sugerencia de los autores de la base de datos es utilizar las señales recogidas en el colegio Ohm para llevar a cabo el entrenamiento de los sistemas y las correspondientes al colegio Mont para test.

Cinco lingüistas etiquetaron las grabaciones según once categorías emocionales: aburrimiento, desesperación, enfado, enfático, felicidad, irritación, neutro, ma-

Tabla 2.3: Distribución del consenso en el etiquetado de la base de datos AIBO

Etiqueta	**0 – 0.25**	**0.25 – 0.5**	**0.5 – 0.75**	**0.75–1**	**Total**
Enfado	2	774	433	283	1492
Enfático	0	2600	799	202	3601
Neutro	0	0	1861	9106	10967
Positivo	0	398	323	168	889
Resto	1207	46	14	0	1267
Total	1209	3818	3430	9759	18216

ternal, recriminativo, sorpresa y otro. Sin embargo, a la hora de utilizar la base de datos se suelen reagrupar en cuatro o cinco emociones más generales.

Junto a la etiqueta final se proporciona un indicador del consenso de los etiquetadores respecto a las emociones detectadas en cada una de las grabaciones. Valores altos indican mayor consenso entre los etiquetadores, mientras que valores bajos indican mayor incertidumbre en la emoción de la frase analizada.

La Tabla 2.3 presenta la distribución de frases grabadas en función de los valores del indicador para las emociones agrupadas que componen la base de datos. Se puede observar cómo un elevado número de frases (en torno al 27 % de la base de datos) tienen un valor del indicador menor de 0.5, gran parte de las cuales (en torno al 6 % de la base de datos) tienen el indicador con valor cero. Puede comprobarse además que el número de ejemplos disponible para cada emoción está fuertemente desequilibrado, estando la mayoría de las señales etiquetadas como neutras.

El audio proporcionado en esta base de datos, el formato de las grabaciones es PCM, mono, con 16 bits de resolución y 16 kHz de frecuencia de muestreo.

2.1.6 NIST i-vector challenge 2013-2014

La base de datos proporcionada para la realización del i-vector challenge 2013-2014 [50] consta de una parte de desarrollo para la creación de los sistemas y una de test reservada para llevar a cabo la evaluación de los mismos. Los locutores que aparecen en ambas partes son diferentes.

A diferencia del resto de bases de datos recogidas en esta sección, el material se entrega en este caso en forma de i-vectors, generados a partir de grabaciones de voz conversacional telefónica utilizadas en anteriores evaluaciones organizadas por el NIST entre 2004 y 2012. Cada i-vector tiene un tamaño definido de 600 componentes. Junto con cada i-vector, se proporciona además información sobre la duración del segmento de voz que ha sido utilizado para calcularlo.

La parte de desarrollo de la base de datos consta de una gran cantidad de i-vectors sin etiquetar, obtenidos a partir de segmentos de voz telefónica de locutores no definidos. La parte de test está formada por un set de 5 i-vectors utilizados para modelar cada uno de los 1.306 locutores a reconocer posteriormente (6.530 i-vectors en total) y un segundo set de 9.634 i-vectors utilizados para llevar a cabo la evaluación de los sistemas (se debe evaluar la presencia de cada locutor objetivo o "*target*", modelado previamente con el material del primer set, en el audio representado por cada i-vector del segundo set).

2.1.7 AMI Meeting Corpus

La base de datos AMI Meeting [21] consta de alrededor de 100 horas de grabaciones diferentes de reuniones. Fue desarrollada a través de fondos europeos (AMI project, FP6-506811) por un consorcio de 15 grupos multidisciplinares dedicado al desarrollo de tecnologías que favorezcan la interacción entre grupos.

Podemos encontrar dos tipos de reuniones diferentes. "Non-Scenario", en las que las reuniones se dan de forma natural y "Scenario", en las que se plantea una situación artificial antes de llevar a cabo las grabaciones. En ambos casos aparecen generalmente 4 participantes en cada grabación.

Todas las grabaciones cuentan con micrófonos próximos y distantes hasta un total de 24 canales para cada reunión. Dichas reuniones se realizaron en inglés en tres habitaciones diferentes con propiedades acústicas distintas, e incluyen en su mayoría hablantes no nativos.

El audio de las grabaciones se reduce desde 48 kHz a 16 kHz, y se entrega en formato PCM (24 archivos para cada reunión, 1 por canal de audio). Adicionalmente se codifica el audio en formato RealMedia para llevar a cabo la transmisión desde el servidor de archivos de medios.

2.1.8 ICSI Meeting Corpus

La base de datos ICSI Meeting [70] es una colección de 75 reuniones grabadas en el Instituto Internacional de Ciencias Informáticas en Berkeley entre 2000 y 2002.

En este caso, todas las reuniones se llevan a cabo de forma natural ("Non-Scenario") entre diversos grupos de trabajo del Instituto, incluyendo el propio grupo de trabajo del proyecto ICSI Meeting.

Las grabaciones tienen una duración de entre 17 y 103 minutos, generalmente en torno a 1 hora de grabación y cuentan con un total de 53 locutores diferentes, con entre 3 y 10 participantes por reunión y un promedio de 6.

El audio de las reuniones contiene grabaciones simultáneas multicanal, hasta un total de 16 canales diferentes para cada reunión, incluyendo múltiples micrófonos próximos y 6 micrófonos de mesa. Se proporciona en formato PCM (un archivo por canal), big-endian, con una resolución de 16 bits y una frecuencia de 16 kHz.

2.2 Bases de datos desarrolladas

En los últimos años, se han realizado importantes esfuerzos para recopilar y transcribir varias bases de datos, tales como la ISL audio [18], o las ya citadas ICSI meeting [70] y AMI meeting [21] en el entorno de reuniones de trabajo, y las bases de datos COST278 [146], la DiSCo German [12] o la 3/24 catalana [158] descrita anteriormente, en el entorno de difusión de audio.

Estas bases de datos de diarización son en su mayoría monolingües. Aquellas que incluyen habla en más de un idioma, como la base de datos COST278, presentan diferentes locutores para los distintos idiomas.

Existen bases de datos multilingües que incluyen locutores con intervenciones en más de un idioma, sin embargo, no han sido diseñadas para llevar a cabo tareas de diarización de audio, sino para estudiar la alternancia de código entre diferentes pares de idiomas [53], realizar tareas de reconocimiento del habla y del lenguaje en lenguas no nativas [67], o para evaluar sistemas de reconocimiento de locutor en entornos multilingües [27].

Como se ha comentado anteriormente, uno de los objetivos principales de esta tesis ha sido la recopilación y grabación de nuevas bases de datos de voz que per-

mitan el desarrollo y la implementación de sistemas de diarización multilingües en los dos principales ámbitos de aplicación, reuniones de trabajo y difusión de audio. A continuación se describen las bases de datos generadas, Ahonews en el entorno de difusión de audio y Ahomeetings en el entorno de las reuniones de trabajo.

2.2.1 Ahonews

El material de audio de esta base de datos de difusión de noticias fue proporcionado por el canal de televisión ETB, perteneciente a la red de televisión pública vasca. Consiste en una colección de grabaciones de noticias en castellano y euskera correspondientes a emisiones de noticias del año 2010. Junto con la voz de los periodistas que narran las noticias, los archivos de audio incluyen entrevistas, música, ruido y en algún caso traducciones simultáneas sobre el audio original [142].

Por cada archivo de audio, ETB proporcionó además un fichero de texto con información sobre el periodista presente en la grabación, el tema abordado y la transcripción de la propia noticia. No obstante, algunos de estos ficheros proporcionaban información demasiado imprecisa, mientras que otros se encontraban vacíos.

Para la creación de la base de datos, se han utilizado los audios de noticias correspondientes a las dos primeras semanas de grabación. Cada archivo de audio ha sido segmentado de forma manual. A continuación, ha sido etiquetado y almacenado de acuerdo a la identidad del periodista principal y a la lengua presentes en el audio, facilitando de esta manera el acceso a diferentes archivos del mismo locutor en dos idiomas diferentes. Los archivos que no disponen de información sobre el locutor se han organizado en base al idioma utilizado únicamente.

El formato de etiquetas de Wavesurfer [134] ha sido seleccionado para las transcripciones proporcionadas. Por cada grabación, se adjunta un archivo de texto con extensión .lab que contiene marcas de tiempo para diferentes eventos acústicos, tales como cambios de turno, silencios, determinadas condiciones de ruido fondo, grabaciones con voz superpuesta, etc.

Un total de 449 archivos han sido etiquetados de forma manual, 321 de los cuales pertenecen a un locutor conocido (177 en castellano y 144 en euskera). 128 archivos pertenecen por tanto a un locutor sin identificar (86 en castellano y 42 en

Tabla 2.4: Características principales de la base de datos Ahonews

Base de datos	Ahonews
Idioma	Castellano & Euskera
Propósito	Diarización de noticias
Número de locutores	46 mujeres & 27 hombres
(Grabaciones en ambos idiomas)	31 mujeres & 17 hombres
Tamaño	8 horas
Número de locutores por grabación	1-5
Cambios de turno por grabación	1-14

euskera). El número de archivos por locutor varía entre 1 y 14 considerando los archivos disponibles en ambos idiomas.

En la Tabla 2.4 se recogen las principales características de la base de datos Ahonews. Las primeras cuatro filas contienen las características generales de la base de datos, que incluyen los diferentes idiomas utilizados por los locutores, la distribución por género de dichos locutores (en detalle los locutores con grabaciones en ambos idiomas), el tamaño de la base de datos y su propósito principal, que en este caso hace referencia al ámbito de aplicación. Las dos últimas filas contienen el número mínimo y máximo de locutores presentes en cada grabación y los cambios de locutor que se pueden encontrar en los archivos de la base de datos.

Como se ha comentado anteriormente, los archivos pertenecientes a la base de datos Ahonews han sido organizados en función del locutor principal y el idioma de la noticia. De esta forma, se facilita la generación de sesiones broadcast artificiales en las que el número de locutores y el idioma utilizado pueden ser controlados. Con esta idea en mente, parte de los archivos etiquetados han sido concatenados con el fin de crear tres sesiones artificiales con diferentes características acústicas, ninguna de las cuales incluye habla solapada.

- *Ahonews_1*: Con una duración de 20 minutos. Incluye voz de 9 locutores diferentes con largas intervenciones en condiciones de bajo nivel de ruido.

- *Ahonews_2*: Duración total de 25 minutos. Con 40 locutores diferentes que realizan intervenciones de corta duración, incluyendo además segmentos con

ruido de fondo y música.

- *Ahonews_3*: 14 minutos de duración. Con 4 locutores diferentes y sólo dos intervenciones por locutor, una en castellano y otra en euskera. De forma similar a la sesión Ahonews_1, presenta largas intervenciones en condiciones de bajo nivel de ruido.

Los archivos de audio de la base de datos se proporcionan en formato Windows PCM (WAV), con una resolución de 16 bits y frecuencia de muestreo de 48 kHz. Todas los archivos son mono y tienen una longitud aproximada de 2 minutos.

2.2.2 **Ahomeetings**

Esta base de datos ha sido grabada en una sala de reuniones perteneciente al laboratorio Aholab durante el año 2013. Consiste en una serie de sesiones independientes en las que diferentes locutores presentan y discuten diversos temas, principalmente en español, además de ciertas intervenciones en euskera [142].

Para llevar a cabo la captura del audio se ha utilizado un sensor Kinect. El sensor Kinect incluye un array lineal de cuatro micrófonos, que permite el procesamiento avanzado de señales, incluyendo localización de la fuente acústica, cancelación de eco o supresión de ruido. Para la creación de la base de datos, se han registrado las señales de los cuatro micrófonos con una velocidad de muestreo de 16 kHz y 16 bits de resolución. Cada sesión tiene una longitud aproximada de 30 minutos y se proporciona en forma de archivo cuadrafónico.

La base de datos ha sido obtenida a lo largo de varios meses. Por ello, con el fin de mantener las características acústicas de las grabaciones, se ha fijado la posición de los locutores presentes en la sala, por lo que todos ellos ocupan deliberadamente el mismo lugar en cada sesión. La posición del sensor Kinect también se ha mantenido constante y los niveles de señal se han comprobado al inicio de cada grabación. No obstante, no todos los locutores están presentes en todas las sesiones, por lo que el número de locutores por sesión varía entre 5 y 10.

En cuanto al etiquetado de las señales, se ha aplicado un proceso de segmentación automática para localizar las intervenciones de cada locutor [87]. A continuación, se ha realizado una revisión manual y se han incluido anotaciones sobre

diferentes eventos acústicos. El sistema de segmentación automática aplicado es capaz de asignar una única etiqueta de locutor en cada instante de tiempo, por lo que los segmentos con voz solapada de más de un locutor han sido identificados durante el proceso de etiquetado manual posterior.

La Tabla 2.5 resume las características principales de la base de datos Ahomeetings. En esta base de datos, como es habitual en el ámbito de las reuniones de trabajo, se dispone de un menor número de locutores diferentes, con mayor interacción entre ellos, que se refleja en un mayor número cambios de turno respecto a la base de datos Ahonews.

Tabla 2.5: Características principales de la base de datos Ahomeetings

Base de datos	**Ahomeetings**
Idioma	Castellano & Euskera
Propósito	Diarización de reuniones
Distribución por género	2 mujeres & 8 hombres
Tamaño	4.5 horas
Número de locutores por sesión	5-10
Cambios de turno por sesión	200-701

Es más fácil conseguir el resultado deseado en piezas cortas.

Gustav Mahler

CAPÍTULO

3

Segmentación de audio

Se denomina segmentación automática al proceso de dividir un archivo determinado en secciones homogéneas de acuerdo con su contenido. Dependiendo de la aplicación final, el objetivo de dicho proceso de segmentación de audio puede ser muy diverso: diferenciar segmentos de voz del ruido o la música [85], separar voz masculina y voz femenina [100], identificar los segmentos correspondientes a distintos locutores [96], etc.

La segmentación automática de audio tiene muchas aplicaciones y por lo general, se utiliza a modo de tratamiento previo del audio para mejorar el rendimiento de sistemas desarrollados en otras áreas del procesado de la voz, como el reconocimiento automático del habla [121], la verificación o el reconocimiento de locutores [115], o la indexación de audio y recuperación de información [94].

Los sistemas de segmentación de audio se pueden clasificar en distintos grupos en función de la técnica de segmentación aplicada. Tradicionalmente, la segmentación de audio se ha llevado a cabo mediante dos métodos principalmente:

- *Segmentación basada en distancia (Metric based segmentation)*: estos métodos hacen uso de una medida de distancia acústica para evaluar la similitud entre dos ventanas adyacentes que se van desplazando a lo largo de un archivo audio. En primer lugar, se identifican los puntos de cambio o fronteras

entre segmentos en base a un determinado umbral, fijado para la medida de distancia entre ventanas elegida. A continuación, se procede al etiquetado de los segmentos resultantes. Las medidas de distancia más ampliamente utilizadas son BIC (Bayesian Information Criterion) [26], Kullback-Leibler [131], Hotlling's T^2 [161] o GLR (Generalized Likelihood Ratio) [93].

- *Segmentación basada en modelos (Model based segmentation)*: estos métodos se basan en la construcción de un conjunto de modelos estadísticos para cada una de las clases acústicas objetivo (voz, ruido, música...). Posteriormente, se divide el audio en segmentos de duración fija y se realiza una clasificación de dichos segmentos utilizando los modelos previamente entrenados, identificando como fronteras los cambios de clase acústica entre segmentos consecutivos. Modelos de mezclas gaussianas [103], modelos ocultos de Markov [109] o redes neuronales [60] son algunos ejemplos de las técnicas más comunes de segmentación de audio basada en modelos.

Las técnicas de segmentación basada en modelos requieren de información previa de las clases acústicas involucradas. Por el contrario, los métodos basados en distancia, utilizados habitualmente en tareas de segmentación de locutores, no hacen uso de ningún tipo de conocimiento previo. En algunos casos, es posible realizar una primera segmentación mediante medidas de distancia para identificar posibles puntos de cambio, para, a continuación, generar modelos a partir de los segmentos acústicos encontrados y llevar a cabo una segunda segmentación que permita refinar la posición de dichas fronteras.

Al referirnos a segmentación de audio, el uso de técnicas basadas en modelos suele ser el criterio más habitual, ya que por lo general, se dispone de muestras de audio de las distintas clases acústicas. Este capítulo se centra por tanto, en las técnicas basadas en el modelado y la posterior clasificación del audio. En primer lugar, se recogen los procesos y métodos involucrados en la clasificación de audio encontrados en la literatura. A continuación, se presentan las métricas y campañas dedicadas a la evaluación los sistemas desarrollados. Por último, se describen las técnicas propuestas para la mejora de la segmentación y los resultados obtenidos en distintas campañas de evaluación. Los métodos basados en distancia se retomarán en el capítulo 6, dedicado a la segmentación de locutores.

3.1 Estado del arte

3.1.1 Parametrización de la señal de audio

Se denomina parametrización al proceso de extracción de las características que definen la señal de voz, y contienen la información más relevante de la señal.

Dado que la voz no es una señal estacionaria, sus características varían con el tiempo, por lo que se deben extraer nuevos parámetros periódicamente. Generalmente, se utiliza para ello un análisis por ventana deslizante, mediante el cual se parametriza un único segmento de la señal de forma individual. A continuación se desplaza la ventana y se repite el proceso para cada uno de los segmentos que forman la señal de voz [156].

En procesado es habitual suponer la señal de voz cuasi-estacionaria, manteniéndose sus características constantes en segmentos de corta duración (generalmente se utilizan ventanas de duración 10-20 ms). Para evitar que se pierdan las muestras situadas en los bordes de los segmentos, a menudo se utiliza solapamiento entre las ventanas que definen dichos segmentos.

Como resultado de la parametrización, se obtiene un conjunto de parámetros o características denominados vectores de parámetros, uno por cada trama, que serán entregados al siguiente bloque del sistema. A continuación se describen las técnicas de parametrización más utilizadas en el estado del arte.

3.1.1.1 Parametrización MFCC

La técnica más utilizada en procesado de la voz, y más concretamente en el ámbito de la diarización, es la parametrización MFCC (Mel Frequency Cepstral Coefficients) [130]. La figura 3.1 muestra el proceso de parametrización en este caso.

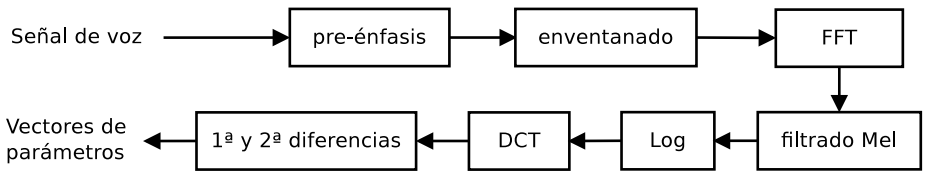


Figura 3.1: Diagrama del proceso de parametrización MFCC

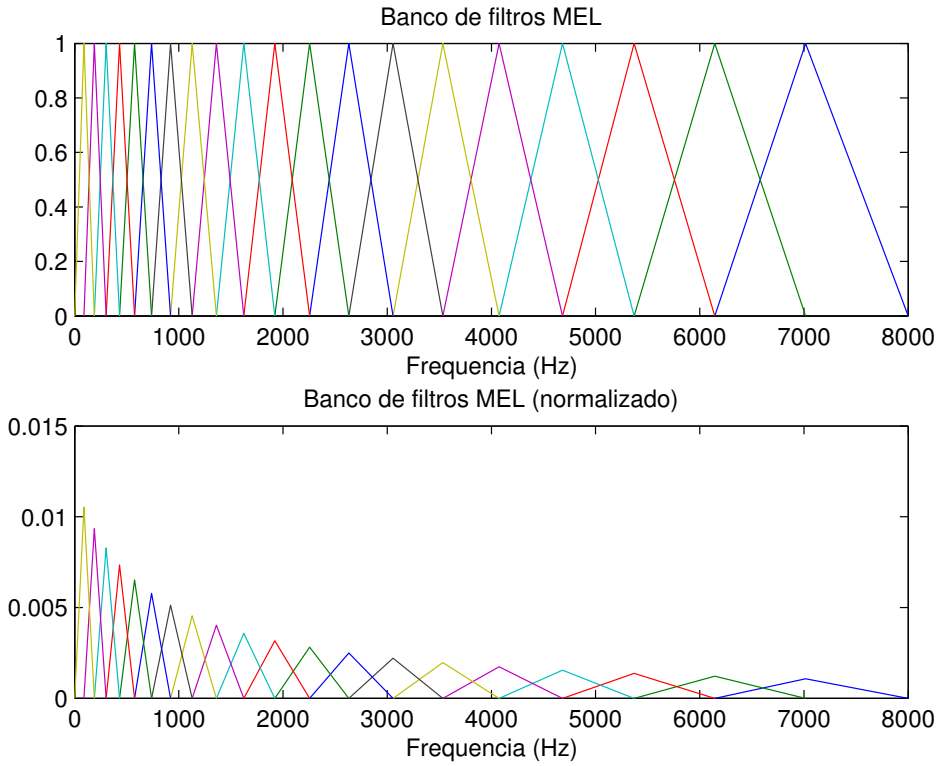


Figura 3.2: Ejemplo de banco de filtros en escala Mel (con y sin normalización)

En primer lugar, la señal de audio pasa por un filtro de preénfasis para eliminar el efecto de radiación de los labios. A continuación, se enventana la señal y, para cada trama definida por la ventana de análisis, se obtiene la transformada discreta de Fourier y se calcula la energía en cada filtro de un banco de filtros en escala Mel.

La figura 3.2 muestra un banco de filtros Mel antes y después de la normalización. Estos filtros tienen forma triangular, y su distribución es no uniforme. En frecuencias bajas los filtros mantienen una anchura constante, mientras que en las frecuencias altas su anchura aumenta exponencialmente. De esta forma, se intenta emular el efecto de enmascaramiento frecuencial que tiene lugar en el oído.

La forma logarítmica comprime los elementos de los vectores, de forma que los coeficientes obtenidos se aproximen a una distribución gausiana. A continuación, se aplica una transformada discreta del coseno o DCT (Discrete Cosine Transform) para decorrelar sus componentes entre sí.

Generalmente, las primeras y segundas diferencias de los elementos (coeficientes de velocidad y aceleración) se añaden a los vectores de parámetros con el fin de retener información acerca de la variación temporal de los mismos.

3.1.1.2 **Parametrización LPCC**

Menos habitual que la parametrización MFCC, destaca el grado de utilización de la parametrización LPCC (Linear Prediction Cepstral Coefficients) [108]. La figura 3.3 describe el proceso de parametrización en este caso.

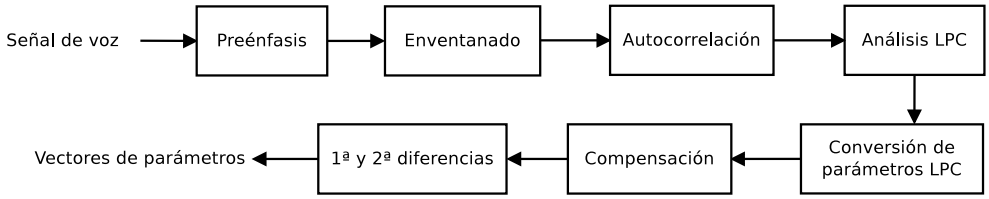


Figura 3.3: Diagrama del proceso de parametrización LPCC

Al igual que en el caso de los MFCC, la señal se pasa por el filtro de preénfasis y se enventana. A continuación, se calcula la autocorrelación para cada una de las tramas y se realiza un análisis LPC. De esta forma los coeficientes de autocorrelación se transforman en un conjunto de parámetros LPC. Para realizar este proceso se utiliza habitualmente el método de Levinson-Durbin.

A partir de los parámetros LPC, a_k, y siendo p el orden del análisis LPC, se obtienen los coeficientes cepstrales LPCC, c_n, mediante el siguiente procedimiento recursivo [48]:

$$c_n = \begin{cases} -a_1, \text{ si } n = 1 \\ -a_n - \sum_{m=1}^{n-1}(1 - \frac{m}{n})a_m c_{n-m}, \text{ si } 1 < n < p \\ -\sum_{m=1}^{p}(1 - \frac{m}{n})a_m c_{n-m}, \text{ si } p < n \end{cases} \tag{3.1}$$

El bloque de compensación trata de minimizar tanto la sensibilidad de los coeficientes cepstrales de más bajo orden a la pendiente de la envolvente espectral,

como la sensibilidad de los coeficientes de más alto orden al ruido. Generalmente se lleva a cabo mediante un liftering paso banda de la forma [73]:

$$c'_n = \left(1 + \frac{L}{2}\sin\left(\frac{\pi n}{L}\right)\right) c_n \tag{3.2}$$

donde L representa el número de coeficientes cepstrales. Estos coeficientes cepstrales constituyen un conjunto muy robusto para tareas de reconocimiento de voz.

De nuevo en este caso, generalmente se añaden las primeras y segundas diferencias de los vectores de parámetros para retener información acerca de la variación temporal de los mismos.

3.1.1.3 **Análisis prosódico**

La mayor parte de los sistemas de diarización o reconocimiento de locutor presentes en el estado del arte hacen uso de la extracción de parámetros de la señal de voz a partir del análisis a corto plazo (tanto parametrización MFCC como LPCC). Sin embargo, este tipo de parametrización no utiliza la información a largo plazo o long-term, que permitiría una mejor caracterización de la señal. Un ejemplo de parametrización a largo plazo lo encontramos en la prosodia.

La información prosódica, basada en el pitch y el contorno de la energía de la voz, ha sido utilizada con anterioridad en trabajos de reconocimiento de locutor [7] [91]. Sin embargo, su uso no se ha visto extendido debido principalmente a la relativa dificultad del cálculo del pitch, así como a la alta carga computacional que conlleva. Los avances en los sistemas de procesado de señal y la significativa mejora en los algoritmos de extracción del pitch han favorecido el retorno al estudio de la prosodia en el ámbito del reconocimiento en los últimos años.

Al contrario que en técnicas de análisis short-term (analizadas anteriormente), es necesario utilizar varias tramas para llevar a cabo la extracción de la información prosódica (cálculo del pitch, energía y las primeras diferencias).

Por otro lado, la información prosódica presenta ciertas ventajas respecto de la espectral, como la robustez frente a efectos de distorsión de canal y al ruido.

3.1.2 Modelado y clasificación de audio

Una vez obtenidos los vectores de parámetros que contienen la información que define la señal de voz, es necesario crear modelos capaces de representar las distintas clases presentes en el audio (fonemas, locutores, eventos acústicos...). Cada uno de estos modelos debe representar por tanto, una forma de clasificar futuros vectores de parámetros, generalmente, con un determinado grado de incertidumbre.

En esta sección se describen las técnicas de modelado y clasificación de audio más ampliamente recogidas en la literatura, como son el modelado HMM-GMM, las redes neuronales y las máquinas de vectores de soporte.

3.1.2.1 HMM-GMM

Un modelo de Markov es una máquina de estados finitos, que cambia de estado en cada unidad de tiempo. En el instante t, al entrar el modelo en el estado S_i, se genera un vector de observaciones o_t a partir de la densidad de probabilidad $b_i(o_t)$. En el instante de tiempo $t+1$, el modelo pasa al estado S_j en función de la probabilidad de transición entre los estados i y j, que está gobernada por la probabilidad discreta a_{ij}. En el caso $i = j$, el modelo permanece en el mismo estado [155].

La figura 3.4 muestra un ejemplo con un modelo de 5 estados. En este caso, solo se permite el paso de estados de izquierda a derecha (left-to-right) sin posibilidad de saltar ningún estado (no-skip).

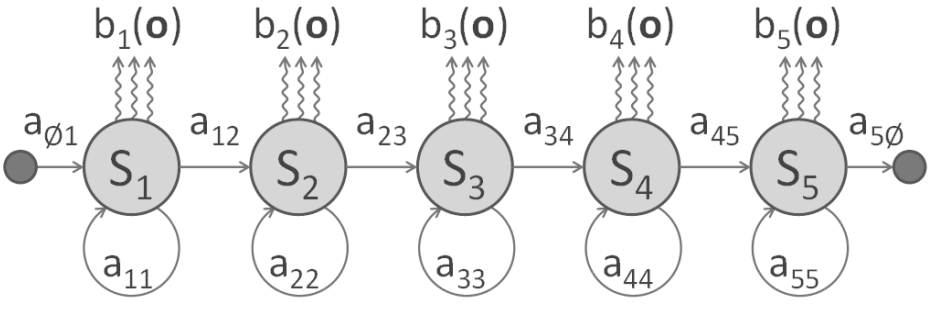


Figura 3.4: Representación gráfica de un HMM de 5 estados de izquierda a derecha (left-to-right) sin posibilidad de saltar ningún estado (no-skip)

La probabilidad de que una secuencia O sea generada por el modelo M mientras recorre la sucesión de estados X se calcula simplemente como el producto de las probabilidades de transición y las probabilidades de emisión. Si suponemos la sucesión de estados $X = 1, 2, 2, 3, 4, 5$ para generar la secuencia $o_1 - o_6$ tenemos:

$$P(O, X|M) = b_1(o_1)a_{12}b_2(o_2)a_{22}b_2(o_3)a_{23}b_3(o_4)a_{34}b_4(o_5)a_{45}b_5(o_6) \tag{3.3}$$

En la práctica sólo la secuencia de observaciones O es conocida, mientras que la sucesión de estados se mantiene oculta, por ello estos modelos reciben el nombre de modelos ocultos de Markov.

Para aplicar este mecanismo al procesado de la señal de voz, deben relacionarse los vectores de observación del modelo con los vectores resultantes de la parametrización de la propia señal. Cada modelo de Markov modelará una cierta unidad de voz (fonema, trifonema, palabra) y en la etapa de entrenamiento se calcularán las probabilidades de transición a_{ij} y las probabilidades de emisión $b_i(o_t)$ que favorezcan la probabilidad $P(O|M)$ de que el modelo emita la secuencia de vectores de observación de esa unidad de voz. Dicha secuencia de vectores de observación se obtendrá a partir de grabaciones de ejemplo de la unidad de voz a modelar.

Durante la etapa de reconocimiento, la secuencia de vectores de observación será conocida, la establecida por los vectores obtenidos tras llevar a cabo la parametrización de la señal de voz a reconocer. Por cada modelo M definido, se calcula la probabilidad $P(O|M)$ de que esa secuencia haya sido generada por ese modelo. La unidad de voz cuyo modelo ofrezca mayor probabilidad de generar esa secuencia de vectores se toma como salida del reconocimiento.

Es habitual que las probabilidades de emisión $b_i(o_t)$, que dan lugar a los vectores en cada estado del modelo, sean definidas mediante distribuciones continuas, siendo la distribución gaussiana la más utilizada, ya que simplifica en gran medida el proceso de entrenamiento [156]. Sin embargo, la distribución real de los vectores de observación rara vez se ajusta a una única gaussiana, por lo que se suelen utilizar distribuciones de varias componentes gaussianas.

Una mezcla gaussiana es una suma ponderada de K componentes gaussianas individuales que viene dada por la ecuación:

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^{K} p_i \cdot b_i(\mathbf{x}) \tag{3.4}$$

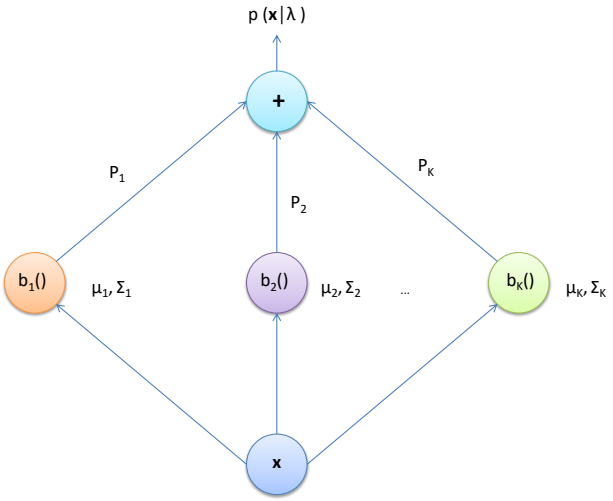


Figura 3.5: Representación gráfica de una mezcla de gaussianas

donde $\mathbf{x}$ es un vector de dimensión D, $b_i(\mathbf{x})$, es la función densidad de probabilidad y p_i, los pesos de cada componente [103]. La función densidad de probabilidad es una gaussiana aleatoria que viene dada por la ecuación:

$$b_i(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma_i}|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu_i})'\boldsymbol{\Sigma_i}^{-1}(\mathbf{x}-\boldsymbol{\mu_i})\} \tag{3.5}$$

donde, $\boldsymbol{\mu_i}$, es el vector de medias y $\boldsymbol{\Sigma_i}$ la matriz de covarianzas. Los pesos de la mezcla deben satisfacer $\Sigma_{i=1}^{K}\, p_i = 1$. Se puede observar en la figura 3.5 un ejemplo de la representación gráfica de una mezcla de gaussianas.

Una mezcla de gaussianas está completamente caracterizada, por tanto, por los vectores de medias, las matrices de covarianza y los pesos de la mezcla de cada una de las componentes. Generalmente, se representan con la notación:

$$\lambda = \{p_i, \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}\}\ i = 1, ..., K \tag{3.6}$$

Un modelo de mezclas gaussianas (GMM, Gaussian Mixture Model) es, básicamente, un HMM de un único estado con una distribución gaussiana de los vectores de observación. Si bien los modelos ocultos de Markov son adecuados para el reconocimiento de voz y la segmentación de audio, no lo son tanto para el reconocimiento de locutor, donde los vectores de observación pueden tratarse de forma

independiente sin estar condicionados por los temporalmente adyacentes. Cada locutor se modela mediante un GMM, y es caracterizado por su modelo λ.

La mezcla gaussiana puede tomar diferente forma dependiendo de las matrices de covarianza utilizadas. Un modelo GMM puede tener una matriz de covarianza por cada componente gaussiana, una matriz de covarianzas para todas las componentes gaussianas del modelo, o una única matriz compartida por todos los modelos. Además, dicha matriz de covarianzas puede ser completa o diagonal, siendo ésta última la opción más eficiente y habitual.

El entrenamiento de un modelo GMM consiste en la estimación de los parámetros $\lambda = \{p_i, \boldsymbol{\mu_i}, \mathbf{\Sigma_i}\}\ i = 1, ..., K$ a partir de los vectores observación obtenidos de las señales de voz del entrenamiento. Para ello, se utiliza habitualmente un proceso de máxima verosimilitud (Maximum Likelihood, ML) mediante algoritmo de maximización de la esperanza (Expectation-Maximization, EM). Generalmente, se utiliza el algoritmo K-Means (KM) para estimar la inicialización del proceso y reducir el número de iteraciones necesarias para que el algoritmo EM converja.

En la etapa de reconocimiento, dada la secuencia de vectores de observación $O = o_1, ..., o_T$, se calcula para cada modelo GMM definido, $M = 1...I$, el valor de log-verosimilitud correspondiente, que viene dado por la ecuación:

$$L(\lambda_M) = \sum_{t=1}^{T} \log P(o_t|\lambda_M)\ M = 1, ..., I \tag{3.7}$$

La clase (locutor, emoción, evento acústico...) cuyo modelo ofrece mayor log-verosimilitud para la secuencia de vectores se toma como salida del sistema.

Decodificación mediante algoritmo de Viterbi

A la hora de realizar el reconocimiento o la segmentación de un audio basado en modelado HMM-GMM, rara vez se utiliza un enfoque trama a trama, identificando en cada instante el modelo que mejor se ajusta a los datos. En su lugar, la técnica comúnmente utilizada en la mayoría de sistemas es la decodificación del audio mediante el algoritmo de Viterbi.

Este algoritmo tiene por objeto obtener la secuencia de estados óptima a partir de las observaciones $O = o_1, ..., o_T$. Para ello, en cada instante t se evalúa la fiabilidad de cada posible secuencia de estados por medio de una función de coste. En

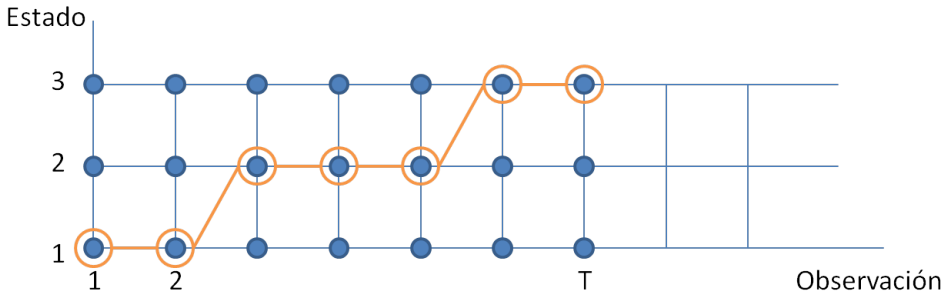


Figura 3.6: Secuencia de estados más probable en un diagrama de Trellis

cada posible estado i y cada instante t se localiza la secuencia de estados más probable de todos los posibles hasta ese punto, identificando en cada caso el camino que presenta un menor coste, que viene dado por:

$$\phi_t(i) = \max_j[\phi_{t-1}(j)a_{ji}]b_i(o_t) \tag{3.8}$$

donde j representa cada uno de los posibles estados en el instante anterior $t-1$, a_{ij} la probabilidad de transición de dicho estado j al actual i y $b_i(o_t)$ la probabilidad de emisión del propio estado i.

El funcionamiento del algoritmo queda reflejado en la figura 3.6, donde podemos observar un diagrama de Trellis. El Trellis enfrenta los estados (eje vertical) y el tiempo (eje horizontal). Los nodos del Trellis representan cada uno de los posibles estados en cada uno de los instantes, que a su vez almacenan la secuencia de estados más probable hasta ese punto.

Se trata de un algoritmo donde el valor obtenido para cada estado en cada instante depende del valor obtenido por el estado de origen en el instante anterior. Cuando el algoritmo llega al final, $t = T$, se recupera recursivamente la secuencia de estados más probable que se toma como salida del sistema.

Adaptación de modelos

Uno de los principales problemas del modelado basado en GMMs radica en la dificultad para obtener la cantidad de audio necesaria para entrenar modelos robustos

de los elementos a reconocer posteriormente (locutores, emociones, eventos acústicos...). Para solucionar este inconveniente, generalmente se recurre a la adaptación de modelos a partir de un Universal Background Model (UBM) [113].

Un UBM es un modelo GMM entrenado para recoger las características que mejor representan el conjunto de los elementos a reconocer. En el caso del reconocimiento de locutores por ejemplo, el modelo debería reunir las características que son independientes del locutor, y que son comunes a todos ellos. Normalmente, la cantidad de material disponible de cada locutor es limitada, sin embargo, es posible obtener un modelo suficientemente robusto a partir de los vectores obtenidos para un grupo formado por varios locutores (habitualmente todos los locutores).

El proceso de adaptación consiste en modificar los distintos parámetros que componen el modelo UBM (medias, varianzas y pesos), para que se ajusten a los de un elemento concreto. En el caso del reconocimiento de locutores, consistirá en aproximar el modelo UBM, que representaría un locutor medio, a las características de un locutor concreto mediante los vectores de observación obtenidos para dicho locutor. En la práctica, es habitual que sean las medias las únicas que sufren modificaciones, debido a que la falta de material disponible a menudo impide que las mejoras obtenidas al modificar varianzas y pesos resulten significativas.

El método de adaptación más habitual consiste en utilizar el conocimiento almacenado en el modelo UBM en un proceso de adaptación bayesiana o MAP (Maximum A Posteriori) para estimar los parámetros de los nuevos modelos. Dados unos vectores de observación $O = o_1, ..., o_T$, y un modelo UBM λ_{UBM} de K componentes gaussianas, los nuevos vectores de medias adaptados μ_i' deben ser obtenidos a partir de las medias del modelo UBM, μ_i. Para ello, en primer lugar se debe calcular la probabilidad de ocupación de la gaussiana iésima de la mezcla para los vectores de observación, que viene dada por:

$$P_{it} = \frac{p_i \cdot b_i(o_t)}{\Sigma_{i=1}^{K} p_i \cdot b_i(o_t)} \tag{3.9}$$

donde b_i representa la función densidad de probabilidad gaussiana definida en la ecuación 3.5. A continuación, se obtienen los estadísticos suficientes definidos por:

$$n_i = \sum_{t=1}^{T} P_{it} \tag{3.10}$$

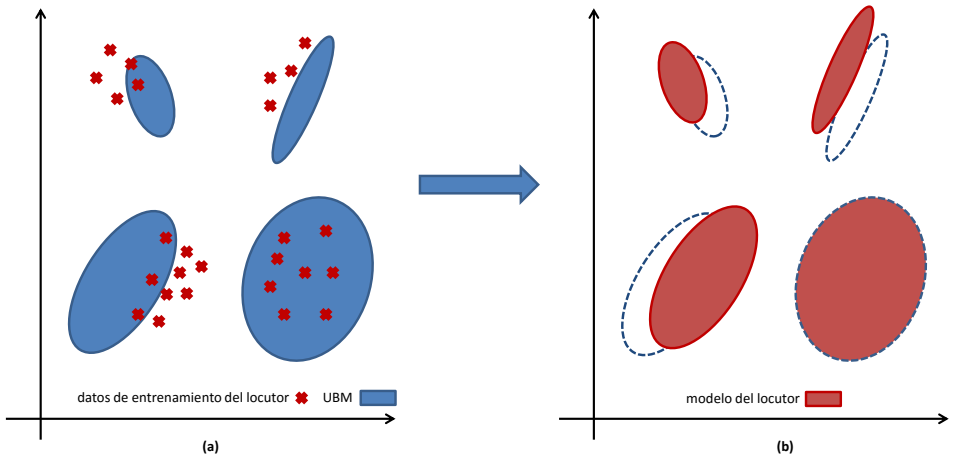


Figura 3.7: Adaptación de gaussianas a partir de un modelo UBM mediante MAP

$$f_i = \sum_{t=1}^{T} P_{it} o_t \tag{3.11}$$

Finalmente, se obtienen los nuevos vectores de medias adaptados a partir de las medias del modelo UBM y los estadísticos suficientes calculados en el paso anterior, mediante la expresión:

$$\mu_i' = \frac{n_i}{n_i + \tau} f_i + \left(1 - \frac{n_i}{n_i + \tau}\right) \mu_i \tag{3.12}$$

donde τ representa el factor de adaptación, que controla la influencia de los vectores de entrenamiento y del modelo UBM en la adaptación de las medias.

El funcionamiento del algoritmo queda reflejado en la figura 3.7, donde se representa la adaptación de gaussianas de dos dimensiones a partir de un modelo UBM, mediante las observaciones de un locutor concreto.

Durante la etapa de reconocimiento, dada la secuencia de vectores de observación $O = o_1, ..., o_T$, se calcula para cada modelo GMM definido, la relación de log-verosimilitud entre dicho modelo y el UBM, dada por la ecuación:

$$L(\lambda_M, \lambda_{UBM}) = \frac{1}{T} \sum_{t=1}^{T} \{\log P(o_t|\lambda_M) - \log P(o_t|\lambda_{UBM})\} \tag{3.13}$$

Una vez más, la clase cuyo modelo ofrece mayor relación de log-verosimilitud para la secuencia de vectores se toma como salida del sistema.

Supervector GMM

Como se ha comentado en la sección anterior, en la práctica sólo las medias de los modelos se adaptan, dado que varianzas y pesos no aportan mejoras significativas. Resulta sencillo, por tanto, pensar que la información relativa a un locutor (o elemento que se desee modelar) se encuentra recogida en su mayor parte en los vectores de medias de su correspondiente modelo y el éxito de los sistemas de reconocimiento basados en adaptación MAP no hace sino refrendar este hecho.

En torno a esta idea, en los últimos años se han obtenido buenos resultados al realizar el modelado concatenando los vectores de medias de los modelos GMM en un vector de gran dimensión, denominado supervector GMM [20] [62] [149].

El esquema básico de este enfoque queda reflejado en la figura 3.8. Se toma en primer lugar un segmento de señal, cuya longitud puede variar entre unos pocos segundos (obteniendo varios vectores para cada una de las clases a modelar) y la totalidad del audio disponible para cada una de las clases a modelar (un único vector conteniendo la información referente a cada clase), siendo el primer enfoque el más utilizado habitualmente. A partir de cada segmento seleccionado se lleva a cabo el entrenamiento de un modelo GMM mediante adaptación MAP de las medias de un modelo UBM previamente entrenado con material correspondiente a todas las clases. Las medias del modelo GMM entrenado, se concatenan a continuación para formar el supervector GMM.

Dado que los supervectores reúnen las características principales de cada clase, es posible considerarlos un nuevo tipo de vector de observación y entrenar un modelo a partir de ellos. En tareas de reconocimiento de locutor, generalmente, se hace uso de un SVM (Support Vector Machine) para llevar a cabo dicho entrenamiento.

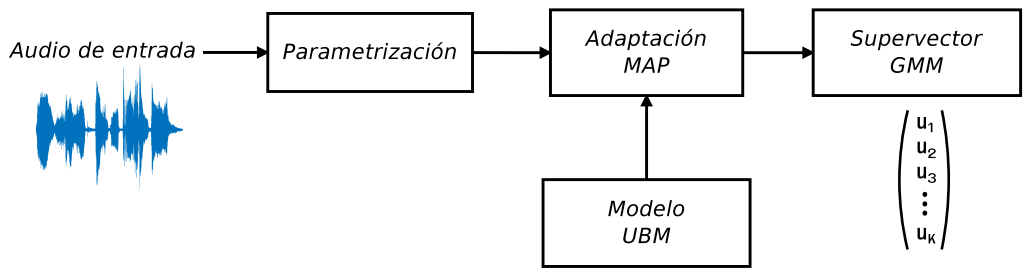


Figura 3.8: Diagrama del proceso de extracción del supervector GMM

3.1.2.2 SVM

Las máquinas de vectores de soporte, propuestas por Vapnik [147] [28], forman parte de los algoritmos de aprendizaje supervisados. Este tipo de clasificadores binarios basan su funcionamiento en la idea de encontrar, a partir de un conjunto de ejemplos, la separación lineal óptima entre muestras positivas y negativas, realizando para ello un mapeo de los datos de entrada a un espacio de elevada dimensión mediante una función no lineal denominada kernel. Se puede observar una representación gráfica de esta idea en la figura 3.9.

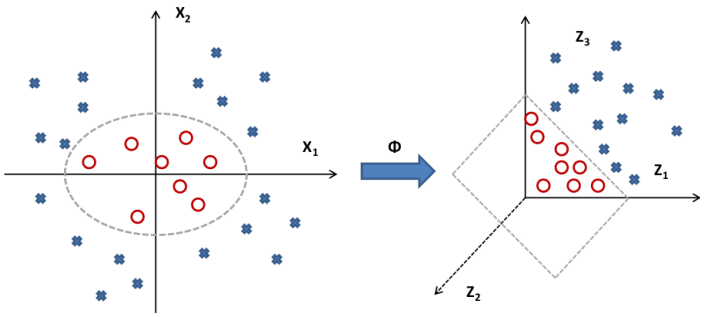


Figura 3.9: Representación de la idea subyacente detrás del modelado SVM

En este nuevo espacio dimensional, se construye una superficie lineal o hiperplano de decisión con propiedades concretas que aseguran una alta capacidad de generalización. En la figura 3.10 se muestra un ejemplo del funcionamiento de un clasificador SVM. De todas las superficies de decisión posibles que separan las dos clases, se busca aquella que maximiza la distancia con los datos de entrenamiento. La mayor probabilidad de error ocurre en la región cercana a la frontera entre clases, por lo que al maximizar el margen se logra minimizar el error.

Suponiendo $C = \{+1, -1\}$ las etiquetas asociadas a cada una de las clases, la frontera se calcula como un hiper-plano $\mathbf{w} \cdot \mathbf{x} + b = 0$, siendo $\mathbf{x}$ un vector de entrada y $(\mathbf{w}, b)$ los parámetros del SVM. La clasificación de un nuevo vector de parámetros $\mathbf{x}$ viene dada por:

$$\hat{c} = sign(\mathbf{w} \cdot \mathbf{x} + b) \tag{3.14}$$

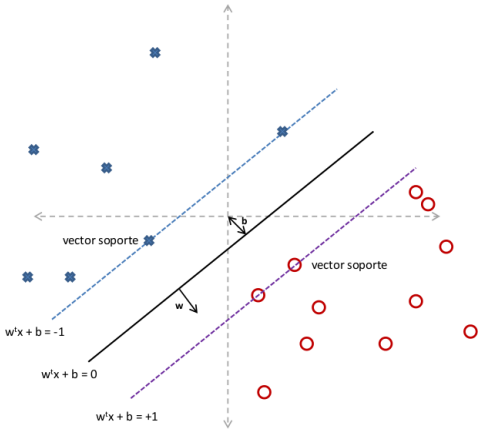


Figura 3.10: Hiper-plano de decisión vectores soporte en el modelado SVM

Teniendo en cuenta la condición de máxima distancia a la frontera, el SVM puede obtenerse mediante la expresión:

$$\mathbf{w} = \sum_{i=1}^{N} \lambda_i y_i \mathbf{x_i} \tag{3.15}$$

donde N representa el número de muestras de entrenamiento, $\mathbf{x_i}$ las propias muestras, y_i las etiquetas de clase correspondientes a esas muestras, $\{+1, -1\}$, y λ_i los pesos asociados a dichas muestras, cuyo cálculo requiere generalmente de un algoritmo de programación cuadrática. Las muestras para las que $\lambda_i \neq 0$ se denominan vectores soporte, y determinan la posición de la frontera.

Como se ha comentado previamente, el funcionamiento del SVM se basa en aplicar una transformación no lineal, Φ, de los vectores de entrada, $\mathbf{x}$, a un espacio de mayor dimensión. Mediante el uso de kernels evitamos definir explícitamente esta transformación Φ, obteniendo una función kernel $k(\mathbf{x_1}, \mathbf{x_2})$ dada por:

$$k(\mathbf{x_1}, \mathbf{x_2}) = \Phi(\mathbf{x_1}) \cdot \Phi(\mathbf{x_2}) \tag{3.16}$$

El mapeo de los vectores llevado a cabo por la función kernel determina por tanto, la separabilidad del problema en ese nuevo espacio y la capacidad de generalización del SVM. Algunas de las funciones kernel más ampliamente utilizadas son RBF (radial basis function), polinómico y sigmoide.

La principal limitación de los clasificadores SVM radica en su discriminación binaria, $\{+1, -1\}$. Para resolver este problema, a menudo se lleva a cabo el entrenamiento de varios SVM para diferentes parejas de clases, determinando la salida final en función de los resultados del conjunto de SVMs entrenados. Los métodos utilizados más frecuentemente son 1vs1, en el que se entrena un SVM para cada par de clases posibles, y 1vsAll, en el que se entrena un SVM para cada clase, capaz de discriminar entre dicha clase y el resto.

En tareas de segmentación de audio, reconocimiento de locutores o reconocimiento de idioma a menudo los clasificadores SVM sustituyen a los clásicos GMM a la hora de llevar a cabo la clasificación del audio [86] [20] [19].

3.1.2.3 **Redes neuronales**

Las redes neuronales (ANN, Artificial Neural Network) toman su nombre debido a que basan su funcionamiento en los procesos que tienen lugar en una neurona biológica [37]. Las redes neuronales no pretenden modelar el comportamiento biológico en sí mismo, sino conseguir un procesamiento de datos adecuado.

La unidad básica de procesamiento se denomina neurona artificial. Cada neurona consta de varias entradas, que se combinan generalmente de forma lineal:

$$in_i = (x_1 w_{i1}) + (x_2 w_{i2}) + ... + (x_n w_{in}) \tag{3.17}$$

donde x_n representa las distintas entradas y w_{in} los pesos de dichas entradas para la neurona i. Al resultado de dicha combinación se aplica una función de transformación que proporciona la salida de la neurona. Algunas funciones de transformación habituales son lineal, sigmoide, tangente hiperbólica o un simple umbral.

Las neuronas se organizan en capas, de forma que las salidas de las neuronas en una capa se unen a las entradas de otras neuronas en la siguiente capa. La primera capa toma como entradas las propias del sistema, y la última capa da lugar a la salida del mismo. El resto de capas se denominan capas ocultas. La figura 3.11 muestra el diagrama de una red neuronal que cuenta con una única capa oculta. Una red neuronal formada por la capa de entrada, dos capas ocultas y la capa de salida es capaz de realizar cualquier mapeado entre las entradas y las salidas [60].

Los parámetros que caracterizan una red neuronal son la topología, que define las conexiones entre las neuronas, y los pesos asignados a cada entrada de las

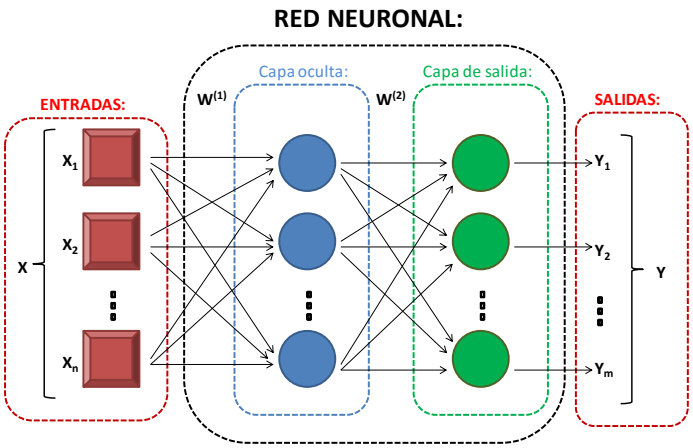


Figura 3.11: Diagrama de la arquitectura básica de una red neuronal

distintas neuronas que componen la red. La topología de la red queda fijada en el diseño del sistema, por lo que no varía durante el entrenamiento. Los pesos deben ser ajustados para conseguir la activación de las salidas adecuadas en cada caso.

La topología determina la capacidad de conexión entre la neuronas de la red, es decir, la forma en la que la salida de una neurona concreta se puede convertir en entrada de otra neurona o de ella misma [69]. Las redes de propagación hacia delante se caracterizan por no permitir que las salidas de las neuronas sean entrada de otras neuronas de la misma capa. En el caso contrario, en el que las salidas se convierten en entradas de neuronas de capas previas (o la misma capa) hablaremos de una red de propagación hacia atrás. Las redes de propagación hacia atrás que presentan además lazos cerrados se denominan recurrentes.

Las redes neuronales más utilizadas en tareas de reconocimiento se denominan MLP (MultiLayer Perceptrons). Se trata de redes con múltiples capas que se propagan únicamente hacia delante, desde la entrada hacia la salida. En cuanto a los algoritmos de aprendizaje, el más extendido es el de back propagation, una generalización del algoritmo LMS (Least Mean Square), y basado por tanto en la minimización del error.

Tras el entrenamiento, la red neuronal se encuentra optimizada para realizar la tarea para la que ha sido diseñada. Cada vez que se presenta un estímulo en la entrada, se propagan las salidas en función de los pesos y las funciones de activación,

y se genera una respuesta en la capa de salida.

En general, un mayor número de neuronas y de capas ocultas supone un mayor margen de la red neuronal para modelar salidas complejas, sin embargo, también se requiere de un mayor número de datos de entrenamiento para estimar todos los parámetros necesarios. En el caso contrario, un número demasiado reducido de neuronas o capas limitará la capacidad de aprendizaje de la red.

Cuando las redes neuronales se utilizan para reconocimiento de voz, reconocimiento de locutores o segmentación, se asignan a la primera capa tantas neuronas como características se han utilizado en la parametrización del audio. En la última capa deberán aparecer tantas neuronas como clases consideradas (número de fonemas, locutores, clases acústicas...). En la etapa de entrenamiento, se calculan los parámetros de la función de activación de cada neurona y los pesos, de forma que cada nodo de salida se active sólo cuando el vector de parámetros a la entrada se corresponda con su respectiva clase. Para llevar a cabo la clasificación de un nuevo vector de entrada, se comprueba cuál de los nodos de la última capa tiene el mayor valor de salida y se asigna al vector la clase correspondiente a ese nodo.

Al igual que en el caso de los clasificadores SVM, las redes neuronales se han convertido en un recurso habitual en sistemas de segmentación de audio [95], reconocimiento automático del habla [58], reconocimiento de emociones [97] o reconocimiento del idioma [110].

3.1.3 Técnicas de reducción de dimensionalidad

Como se ha comentado anteriormente, en los últimos años se han obtenido buenos resultados mediante la concatenación de los vectores de medias de los modelos GMM en un único supervector GMM. Sin embargo, la elevada dimensión de dichos supervectores supone un incremento en el coste computacional, al aumentar el tiempo de entrenamiento de los distintos algoritmos. Además, el número de parámetros irrelevantes generalmente contribuye negativamente en el rendimiento de los sistemas. Por ello, a menudo se hace uso de diferentes técnicas de reducción de la dimensionalidad que permitan identificar el menor número de parámetros necesarios para mantener la información más relevante contenida en los datos.

El análisis factorial es una técnica estadística de reducción de dimensionalidad de los datos cuyo objetivo consiste en expresar los datos observables como una combinación lineal de variables latentes denominadas factores. Una de las aplicaciones del análisis factorial consiste en identificar los factores responsables de la variabilidad de los resultados en un entorno concreto. En los últimos años se han realizado múltiples estudios en el ámbito del reconocimiento de locutor que intentan modelar la variabilidad de locutor y de sesión con buenos resultados en sistemas basados en GMM-UBM. Los métodos más ampliamente utilizados son el análisis factorial conjunto (Joint Factor Analysis, JFA) [74], y el análisis factorial de variabilidad total (Total Variability Front-End Factor Analysis o i-Vectors) [33].

Estos métodos tratan de lidiar con la elevada dimensionalidad de los supervectores de medias, obtenidos a partir de la adaptación MAP de los modelos GMM, proyectándolos a un subespacio más discriminativo de dimensión más reducida. En esta sección se describen brevemente ambos métodos.

3.1.3.1 Joint Factor Analysis

En los últimos años, el Joint Factor Analysis se ha consolidado como estado del arte en técnicas de identificación y verificación de locutores [32] [75] [76]. Este enfoque trata de evitar la variabilidad en los modelos GMM modelando por separado el efecto de la variabilidad entre locutores y la variabilidad de canal.

Supongamos C el número de componentes del modelo UBM y F la dimensión del vector de características acústicas. Nos referiremos con CF a la dimensión del

supervector resultante de la concatenación de los vectores de medias del modelo GMM correspondiente a un segmento de audio.

El JFA [74] supone que un supervector $\mathbf{s}'$ correspondiente a un locutor puede ser descompuesto en una suma de dos supervectores, el supervector de locutor $\mathbf{s}$ y el supervector de canal $\mathbf{c}$ según:

$$\mathbf{s}' = \mathbf{s} + \mathbf{c} \tag{3.18}$$

siendo $\mathbf{s}$ y $\mathbf{c}$ vectores estadísticamente independientes que siguen una distribución normal. Así mismo, se define la descomposición de $\mathbf{s}$ como:

$$\mathbf{s} = \mathbf{m} + \mathbf{V}\mathbf{y} + \mathbf{D}\mathbf{z} \tag{3.19}$$

donde $\mathbf{m}$ representa un supervector de dimensión $CF \times 1$, $\mathbf{V}$ una matriz rectangular de rango bajo, $\mathbf{y}$ un vector que sigue una distribución normal, $\mathbf{D}$ una matriz diagonal de dimensión $CF \times CF$ y $\mathbf{z}$ un vector de dimensión CF que sigue una distribución normal. Las columnas de la matriz $\mathbf{V}$, que representa la variabilidad entre locutores, se denominan eigenvoices, mientras que los elementos del vector $\mathbf{y}$ se conocen como factores de locutor.

Por su parte, el supervector de canal $\mathbf{c}$ se modela como:

$$\mathbf{c} = \mathbf{U}\mathbf{x} \tag{3.20}$$

donde $\mathbf{U}$ representa una matriz rectangular de rango bajo y $\mathbf{x}$ un vector que sigue una distribución normal. Las columnas de la matriz $\mathbf{U}$, que representa la variabilidad de sesión, se denominan eigenchannels, mientras que los componentes del vector $\mathbf{x}$ se conocen como factores de sesión o factores de canal.

El número de eigenvoices y eigenchannels escogido en el diseño del sistema determinarán el tamaño de las matrices $\mathbf{V}$ y $\mathbf{U}$, así como el número de factores de locutor y de canal utilizados en el modelo.

3.1.3.2 **Front-End Factor Analysis**

La técnica de JFA define dos espacios de variabilidad diferentes: el espacio de locutor, definido por la matriz de eigenvoices $\mathbf{V}$ y el espacio de canal definido por la matriz de eigenchannels $\mathbf{U}$. Los experimentos llevados a cabo en [30], sin

embargo, muestran cómo los factores de canal en el JFA, que deberían modelar únicamente los efectos de canal, contenían además información de locutor.

Esto dio lugar a un nuevo enfoque, Front-End Factor Analysis [33], donde se define un único espacio de variabilidad, denominado espacio de variabilidad total o total variability space. Este nuevo espacio contiene de forma simultánea la variabilidad de locutor y de canal.

Dado un segmento de audio, el nuevo supervector GMM dependiente de locutor y de canal se puede reescribir como:

$$\mathbf{m}' = \mathbf{m} + \mathbf{T}\mathbf{w} \tag{3.21}$$

donde $\mathbf{m}$ representa un supervector independiente del locutor y del canal, $\mathbf{T}$ una matriz rectangular de rango bajo y $\mathbf{w}$ un vector que sigue una distribución normal. En este caso, la matriz $\mathbf{T}$ representa la variabilidad entre locutores y sesiones, mientras que los elementos del vector $\mathbf{w}$ definen los factores totales. Estos nuevos vectores reciben el nombre de vectores identidad o i-vectors.

En los últimos años, este nuevo enfoque ha sustituido la técnica de JFA en tareas de reconocimiento y verificación de locutores [82] [123], reconocimiento de idioma [31] [35], o diarización de locutores [132] [129].

3.2 Evaluación en segmentación de audio

3.2.1 Medidas de evaluación

Siguiendo el modelo de las evaluaciones organizadas por el NIST (*National Institute of Standards and Technology*), la métrica más comúnmente utilizada para evaluar distintos sistemas de segmentación de audio es el SER (Tasa de Error de Segmentación o *Segmentation Error Rate*) [98], que se corresponde con la fracción de tiempo de la clase correspondiente (voz, música y ruido en este caso) que no ha sido correctamente asignada. En las zonas de solapamiento entre clases la duración del segmento se atribuye a todas las clases presentes en el mismo, por lo que un mismo segmento temporal puede ser considerado más de una vez en los cálculos.

Dado un dataset Ω en el que llevar a cabo la evaluación de los sistemas, cada señal de audio se divide en segmentos contiguos comprendidos entre dos cambios de clase y se define el tiempo de error de segmentación en ese segmento como:

$$\Xi(n) = T(n)[\text{máx}(N_{ref}(n), N_{sis}(n)) - N_{correct}(n)] \tag{3.22}$$

donde $T(n)$ representa la duración del segmento n, $N_{ref}(n)$ el número de clases presentes en el segmento n definidas en la referencia, $N_{sis}(n)$ el número de clases asignadas por el sistema al segmento n y $N_{correct}(n)$ el número de clases asignadas correctamente por el sistema al segmento n.

A continuación se calcula el error de segmentación como la relación entre el tiempo total de error de segmentación obtenido y la duración total de los segmentos asignados a cada clase en el audio según:

$$SER = \frac{\Sigma_{n\epsilon\Omega}\Xi(n)}{\Sigma_{n\epsilon\Omega}T(n)N_{ref}(n)} \tag{3.23}$$

El SER implica la suma de tres tipos de errores: el porcentaje de tiempo que es asignado a una clase incorrecta (Error de Clase o *Class Error Time*), el porcentaje de tiempo en el que una clase presente en el segmento no ha sido etiquetada (Error de Omisión o *Missed Class Time*) y el porcentaje de tiempo en que se ha etiquetado una clase que no estaba presente en el segmento (Error de Inserción o *False Alarm Time*). Todos estos errores se calculan generalmente mediante las herramientas de evaluación proporcionadas por el propio NIST.

3.2.2 Campañas Albayzin de segmentación de audio

Las campañas competitivas de evaluación son una herramienta adecuada para determinar de manera objetiva la validez de los algoritmos desarrollados. En estas campañas distintos grupos de investigación prueban sus algoritmos sobre una base de datos común, lo que permite comparar el rendimiento de los mismos e identificar las técnicas más adecuadas para cada etapa del sistema.

La Red Temática en Tecnologías del Habla organiza cada dos años las campañas de evaluación Albayzin que evalúan distintos aspectos relacionados con las tecnologías del habla. La segmentación de audio se ha incluido en las tres últimas campañas realizadas, Albayzin 2010 [158], Albayzin 2012 [102] y Albayzin 2014 [22]. En esta sección se describen los objetivos perseguidos en dichas evaluaciones.

Campaña Albayzin de segmentación de audio 2010

La campaña de 2010 persigue la segmentación de audio broadcast, más concretamente de programas de televisión. El objetivo principal es el de asignar a los distintos segmentos etiquetas para indicar la presencia de 5 clases de audio definidos previamente: voz, música, voz con música, voz con ruido y otros.

Campaña Albayzin de segmentación de audio 2012

La campaña de 2012 mantiene como objetivo la segmentación de audio broadcast, principalmente de programas de radio. En esta ocasión, sin embargo, no se definen clases de audio objetivo, sino que se persigue una segmentación multicapa, identificando la presencia de voz, música y ruido en el audio, pudiendo existir solapamiento entre las clases en cualquier instante.

Campaña Albayzin de segmentación de audio 2014

La campaña de 2014 mantiene como objetivo la segmentación multicapa de audio broadcast. En este caso, se proporciona un marco experimental que aumenta la dificultad de las últimas ediciones mediante la combinación de distintas bases de datos, evaluando de esta forma la robustez de los sistemas participantes.

3.3 Mejoras propuestas en segmentación de audio

En esta sección se describen las dos técnicas desarrolladas en esta tesis para la mejora de la segmentación de audio, ambas orientadas al ámbito de difusión de noticias. La primera solución propuesta está basada en el postprocesado de segmentos de voz en busca de música de fondo de bajo nivel, la segunda consiste en la realización de una segmentación más robusta mediante clasificación de i-vectors.

Adicionalmente, se recogen los resultados obtenidos por los sistemas de segmentación de audio desarrollados con motivo de la participación en las campañas de evaluación Albayzin 2012 y 2014, y que implementan las técnicas propuestas.

3.3.1 Postprocesado de segmentos de voz-música

Las grabaciones de radio y televisión presentan a menudo segmentos de voz con música de fondo de bajo nivel difícilmente detectable. La inclusión de un modelo capaz de diferenciar dichos segmentos requiere generalmente una parametrización y un modelado más complejos, lo que supone aumentar considerablemente el tiempo de procesado del audio y los recursos utilizados por el sistema. La alternativa propuesta en este caso consiste en realizar la segmentación del audio en dos etapas.

En primer lugar, se propone realizar una segmentación clásica con modelos entrenados para voz limpia, música, ruido, voz con ruido de fondo, voz con música de fondo, y voz con música y ruido. Estos modelos pueden ser utilizados para realizar una clasificación directa del audio o como entrada en una segmentación mediante el algoritmo de Viterbi, obteniendo resultados adecuados con un reducido número de parámetros y un rápido procesado de las señales.

En la segunda etapa se llevará a cabo el postprocesado de los segmentos de voz limpia encontrados, con el fin de detectar música de fondo de bajo nivel en cada segmento de audio analizado. Para ello, en primer lugar, se deben entrenar un modelo para voz limpia y un modelo para voz con música de fondo a partir de las señales de desarrollo disponibles. A continuación, para cada segmento de voz marcado en la primera etapa, se realiza una nueva segmentación incluyendo únicamente los dos modelos entrenados. En este caso es posible utilizar una parametrización y un modelado más complejos sin comprometer en exceso el tiempo

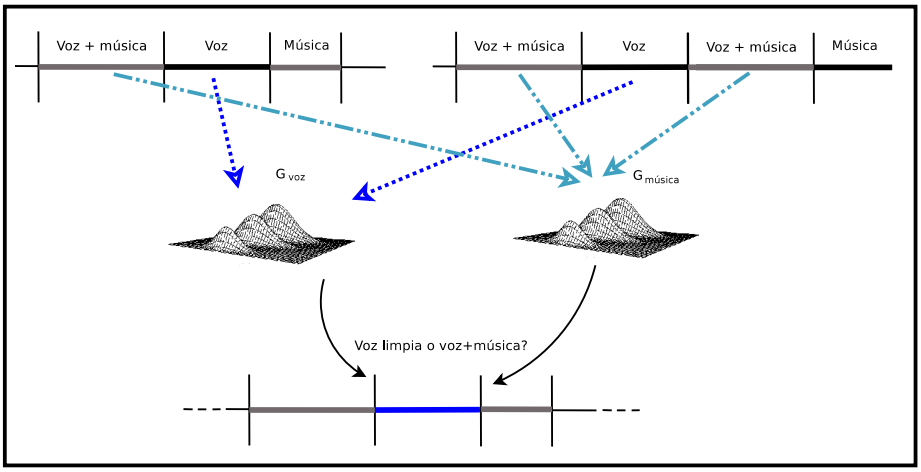


Figura 3.12: Diagrama de la etapa de postprocesado de segmentos de voz propuesta

de procesado necesario, ya que la cantidad de audio a analizar se ha reducido considerablemente en la primera etapa. La figura 3.12 muestra un diagrama del proceso, donde G_{musica} y G_{voz} representan los modelos entrenados con segmentos de audio correspondientes a voz con música de fondo de bajo nivel (solo se utilizan los segmentos con bajo nivel de música en este caso) y voz limpia respectivamente.

Se trata de un método sencillo pero adecuado a las necesidades de este tipo de audio broadcast, donde uno de los principales problemas reside en la detección correcta de los segmentos que contienen música de fondo (en menor medida los segmentos en los que la música es el único elemento presente en el audio).

3.3.1.1 Validación del método propuesto

Como se ha comentado anteriormente, la campaña de evaluación de sistemas de segmentación de audio Albayzin 2012 consistió en la segmentación de audio broadcast. La base de datos utilizada, descrita en el capítulo 2, está formada por unas 20 horas de audio con la siguiente distribución: 22 % de voz limpia, 9 % de música, 31 % de voz con música de fondo, 26 % de voz con ruido de fondo y 12 % de otros. Del total de grabaciones de Aragón Radio disponibles, 32 sesiones se destinaron al desarrollo de los distintos sistemas, mientras que las 72 restantes fueron reservadas por la organización para la realización del test.

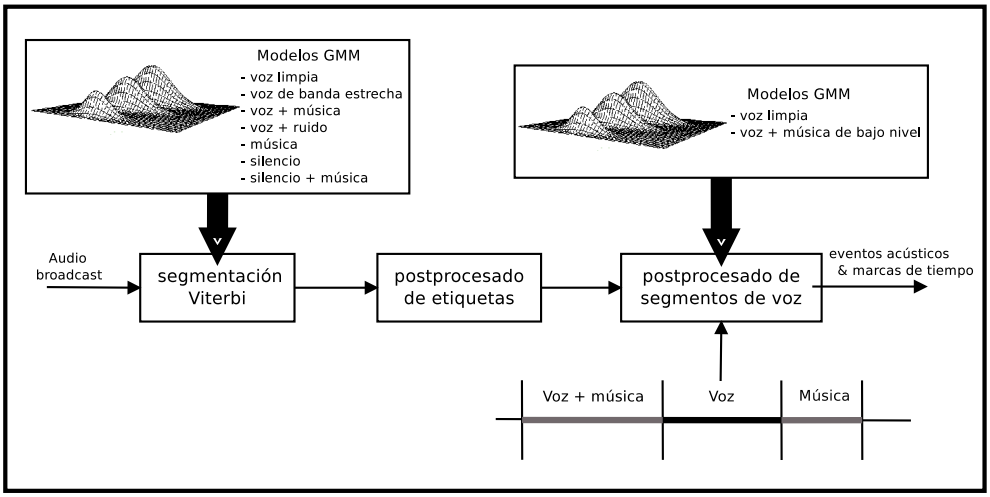


Figura 3.13: Diagrama del sistema de segmentación de audio Albayzin 2012

La figura 3.13 muestra el diagrama del sistema de segmentación de audio desarrollado con motivo de la celebración de la campaña de evaluación Albayzin 2012, y que hace uso de la técnica de postprocesado propuesta [137].

En primer lugar, se entrenaron modelos GMM utilizando mezclas de 1024 gaussianas para voz limpia, música, voz con ruido, voz con música, voz de banda estrecha, silencio y silencio con música de fondo, utilizando aproximadamente 4 minutos de audio (para cada modelo) de las sesiones de entrenamiento de la base de datos. Estos modelos fueron posteriormente utilizados en una segmentación mediante el algoritmo de Viterbi para identificar los segmentos de audio que contienen los distintos eventos acústicos descritos. Para llevar a cabo la clasificación de los segmentos se utilizó en este caso una parametrización construida mediante 6 coeficientes MFCC añadiendo primeras y segundas diferencias.

La finalidad del bloque de postprocesado de etiquetas es refinar los límites de los segmentos, eliminar silencios demasiado cortos y corregir algunos errores menores cometidos en el proceso de segmentación. Así mismo, se unifican los segmentos consecutivos con la misma etiqueta para mejorar el rendimiento de la etapa de postprocesado posterior.

Por último se lleva a cabo el postprocesado de los segmentos de voz limpia encontrados en la etapa anterior. Se entrenaron para ello dos modelos GMM utili-

zando mezclas de 1024 gaussianas para voz limpia, G_{voz} y para voz con música de fondo de bajo nivel, G_{musica}, utilizando 15 minutos de audio (para cada modelo) extraídos de las sesiones de entrenamiento de la base de datos. Cada segmento de voz limpia encontrado es analizado y si se ajusta mejor al modelo G_{musica} que al de voz limpia, la etiqueta de música es añadida a la marca de voz ya existente. En esta etapa, con una menor cantidad de audio para procesar, se utiliza una parametrización más compleja, con 12 coeficientes MFCC añadiendo primeras y segundas diferencias, que permite una mejor diferenciación entre los modelos.

A continuación se muestran los resultados obtenidos por el sistema propuesto en la campaña de evaluación Albayzin 2012. La métrica definida por la organización de la campaña fue el SER, definido previamente en la sección 3.2 como la fracción de tiempo que no ha sido correctamente atribuida a la clase correspondiente (voz, música o ruido). Todos los resultados reflejados en las tablas han sido obtenidos por medio del script de evaluación proporcionado por la organización.

La tabla 3.1 muestra los resultados obtenido por el sistema propuesto en las 32 sesiones de entrenamiento de la base de datos. Se muestran de forma conjunta los resultados obtenidos antes y después de aplicar la etapa de postprocesado de los segmentos de voz limpia. De esta forma, la comparación de los valores de SER obtenidos resulta más sencilla.

Como se muestra en la tabla 3.1, existe una gran variabilidad en los resultados obtenidos en las sesiones, debido principalmente al etiquetado erróneo de la música y el ruido en algunas de las grabaciones. Podemos observar cómo gran parte de los valores obtenidos no presentan cambios significativos al aplicar el postprocesado de los segmentos de voz, por lo que no se han detectado segmentos con música de fondo de bajo nivel en estos casos. En cuanto a las señales que presentan cambios tras aplicar el postprocesado, podemos diferenciar entre las grabaciones en las que se reduce el error de segmentación (2,6,10...) y los casos en los que éste aumenta (11,15,24...). Se observa que existe un mayor número de sesiones en las que el error de segmentación se ve reducido al aplicar la etapa postprocesado propuesta. Así mismo la reducción obtenida es, en general, superior al aumento producido en los casos en los que el funcionamiento postprocesado no resulta adecuado.

En líneas generales, se consigue una reducción relativa del 2.15 % en el error de segmentación en la parte de entrenamiento de la base de datos, lo que apenas

Tabla 3.1: Resultados obtenidos por el sistema de segmentación desarrollado en las sesiones de entrenamiento de la base de datos Albayzin 2012 en términos de SER

Sesión	Viterbi	Postprocesado	Sesión	Viterbi	Postprocesado
1	25.67 %	25.67 %	17	1.61 %	1.61 %
2	39.68 %	12.71 %	18	5.02 %	5.02 %
3	43.99 %	43.99 %	19	28.36 %	28.39 %
4	20.74 %	20.74 %	20	24.71 %	24.60 %
5	28.29 %	28.29 %	21	28.36 %	28.57 %
6	35.82 %	8.95 %	22	2.24 %	2.24 %
7	37.09 %	37.09 %	23	8.92 %	8.92 %
8	3.25 %	3.25 %	24	11.85 %	15.04 %
9	36.25 %	31.12 %	25	31.73 %	33.93 %
10	34.44 %	12.78 %	26	8.75 %	8.75 %
11	18.65 %	19.66 %	27	30.55 %	54.05 %
12	37.84 %	34.97 %	28	13.51 %	12.81 %
13	28.54 %	24.82 %	29	31.32 %	43.29 %
14	6.74 %	6.74 %	30	9.68 %	9.91 %
15	19.60 %	25.60 %	31	16.30 %	16.30 %
16	8.61 %	8.57 %	32	40.09 %	25.23 %
1-32	**21.45 %**	**20.99 %**			

demuestra la validez de la técnica de postprocesado propuesta. Sin embargo, si los distintos eventos acústicos (voz, música y ruido) son evaluados de forma individual, como sucede en las tablas 3.2 y 3.3, resulta evidente el modo en que se ha producido dicha reducción del SER. La tabla 3.2 muestra el error de segmentación obtenido para las clases de voz, música y ruido individualmente antes de aplicar la etapa de postprocesado. Del mismo modo, la tabla 3.3 muestra los mismos resultados tras aplicar el bloque de postprocesado de los segmentos de voz.

Se puede observar en la tabla 3.2 cómo la principal fuente del error de segmentación reside en la detección de ruido y de música, con un Error de Clase obtenido del 42.76 % y del 44.33 % respectivamente, mientras que la clase de voz obtiene buenos resultados con un 5.19 % de Error de Clase.

Tabla 3.2: Detalle del error cometido, en términos de SER, por el sistema de segmentación básico para las clases de voz, música y ruido de forma individual en las sesiones de entrenamiento de la base de datos Albayzin 2012

Tipo de error	Voz	Música	Ruido
Error de Omisión	4.1 %	38.7 %	34.5 %
Error de Inserción	1.1 %	4.1 %	9.9 %
Error de Clase	5.2 %	42.8 %	44.4 %

Tabla 3.3: Detalle del error obtenido, en términos de SER, para las clases de voz, música y ruido de forma individual tras aplicar la etapa de postprocesado en las sesiones de entrenamiento de la base de datos Albayzin 2012

Tipo de error	Voz	Música	Ruido
Error de Omisión	4.1 %	26.7 %	34.5 %
Error de Inserción	1.1 %	4.3 %	9.9 %
Error de Clase	5.2 %	31.0 %	44.4 %

Al aplicar la etapa de postprocesado, es posible identificar segmentos de voz con música de fondo de bajo nivel, lo que se refleja en los resultados obtenidos, recogidos en la tabla 3.3. El postprocesado de los segmentos es capaz de reducir considerablemente el Error de Omisión para la clase de música, con una reducción relativa del 27 %, lo que demuestra la validez de esta etapa de postprocesado para la mejora del etiquetado de esta clase.

En cuanto al tiempo de procesado, la tabla 3.4 presenta el tiempo medio de CPU requerido para analizar las sesiones de entrenamiento de la base de datos. Se puede observar cómo los valores se mantienen por debajo del tiempo real incluyendo la segmentación y el postprocesado (en quad-core 2.27 GHz, 32 GB RAM).

Tabla 3.4: Tiempo de CPU invertido en procesar las sesiones de entrenamiento

Base de datos	Sistema Básico	Postprocesado
5 h. 16 min. 34 s.	2 h. 53 min. 16 s.	33 min. 30 s.

Tabla 3.5: Resultados obtenidos por el sistema de segmentación desarrollado en las sesiones de test de la base de datos Albayzin 2012 en términos de SER

	Sesiones 1-72
Sistema Básico	26.63 %
Postprocesado	**25.78 %**

Tabla 3.6: Detalle del error obtenido, en términos de SER, para las clases de voz, música y ruido de forma individual tras aplicar la etapa de postprocesado en las sesiones de test de la base de datos Albayzin 2012

Tipo de error	Voz	Música	Ruido
Error de Omisión	3.3 %	36.9 %	34.8 %
Error de Inserción	0.9 %	6.7 %	28.2 %
Error de Clase	4.2 %	43.6 %	63.0 %

Resultados similares se consiguieron en las sesiones de test de la base de datos, recogidos en la tabla 3.5. En este caso se consigue una reducción relativa del 3.2 % en el error de segmentación al aplicar la etapa de postprocesado de los segmentos de voz, con un SER final obtenido del 25.78 %.

La tabla 3.6 muestra el error de segmentación obtenido para las clases de voz música y ruido individualmente tras aplicar la etapa de postprocesado propuesta a las señales de test. Se puede observar cómo, a pesar del buen funcionamiento de la etapa de postprocesado de los segmentos de voz limpia, la principal fuente de error se mantiene en la detección de música y de ruido, con un Error de Clase obtenido del 43.6 % y del 63.03 % respectivamente, mientras que se han obtenido resultados muy satisfactorios para la clase de voz con un 4.2 % de Error de Clase. En definitiva, podemos considerar que el sistema constituye una buena alternativa para la extracción de voz en audio broadcast.

En la campaña de evaluación de segmentación de audio Albayzin 2012 tomaron parte 6 sistemas desarrollados por 5 grupos de investigación diferentes, siendo el sistema propuesto el que obtuvo mejor resultado.

3.3.2 Segmentación robusta mediante i-vectors

Los resultados presentados en la sección anterior han demostrado la validez de la técnica de postprocesado de los segmentos de voz para la mejora del etiquetado de la clase de música. Sin embargo, la presencia de esta clase en las bases de datos de segmentación es limitada, y los resultados globales sobre el error de segmentación muestran diferencias poco significativas (recordemos la reducción relativa del 3.2 % conseguida en la base de datos Albayzin 2012).

La figura 3.14 muestra, a modo de ejemplo, el reparto de clases de la base de datos utilizada en la campaña de evaluación de segmentación de audio Albayzin 2014. Se puede observar cómo las clases que contienen voz representan más del 92 % del audio disponible (30.16 % voz limpia, 20.34 % voz con música, 30.64 % voz con ruido y 11.08 % voz con música y ruido).

Las clases que contienen voz dominan claramente la base de datos, por lo que resulta necesaria una mejor clasificación de estos segmentos a la hora de llevar a cabo una segmentación más precisa. La nueva técnica propuesta se basa por tanto en una presegmentación en busca de los segmentos que contienen voz, y un refinamiento posterior de dichos segmentos para determinar la presencia de música y/o ruido de fondo en el audio analizado.

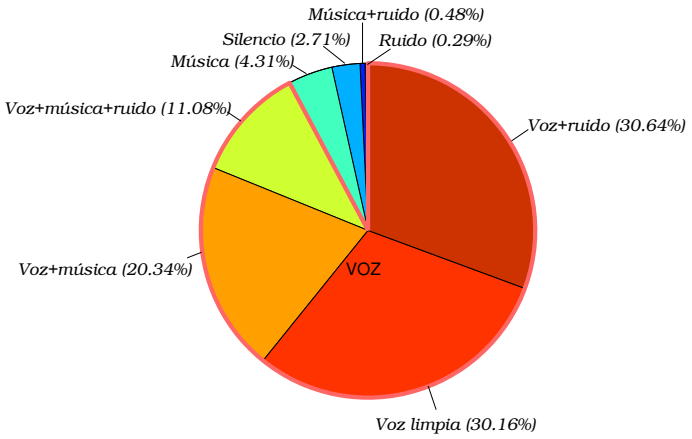


Figura 3.14: Distribución del audio en la base de datos Albayzin 2014

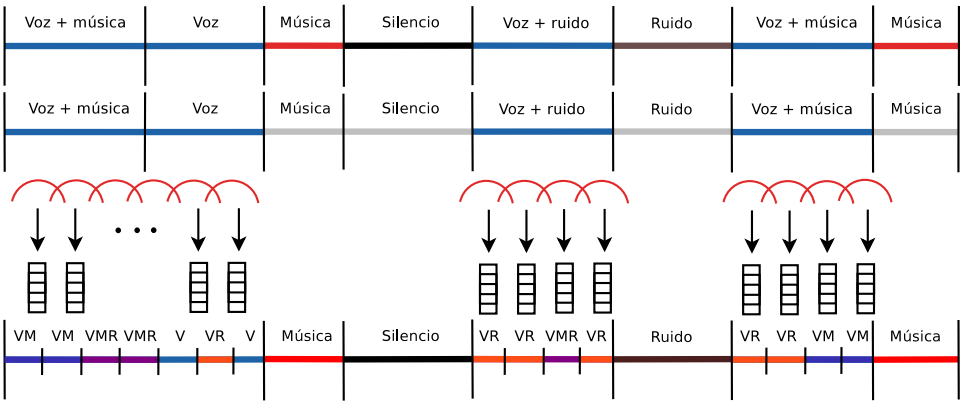


Figura 3.15: Funcionamiento de la técnica de segmentación de audio propuesta

Una sistema tradicional basado en GMM-HMMs constituye una buena alternativa para llevar a cabo la presegmentación de voz/no voz del audio (recordemos el 4.2 % de SER conseguido para la clase de voz en la base de datos Albayzin 2012).

En cuanto a la clasificación de los segmentos de voz (limpia, con música, con ruido...), los resultados obtenidos mediante sistemas basados en GMM-HMMs distan de ser satisfactorios. La utilización de supervectores, entrenados con material de cada una de las clases, se presenta como una alternativa más eficiente. El elevado número de parámetros de dichos vectores, sin embargo, no parece adecuado para la caracterización del fondo de los segmentos de voz, cuya diferenciación reside presumiblemente en un espacio de menor dimensión. Por esta razón, se propone el uso de i-vectors para llevar a cabo la clasificación precisa de los segmentos.

Los sistemas basados en i-vectors han tenido gran relevancia en el campo del reconocimiento de locutor. En los últimos años, su utilización se ha extendido a distintas áreas en tecnologías de la voz, como reconocimiento de emociones [89], reconocimiento de género [150], o adaptación de locutor [52]. En este caso, utilizaremos esta técnica para llevar a cabo una segmentación de audio más robusta.

La figura 3.15 muestra un diagrama de la técnica de mejora de la segmentación de audio propuesta. En primer lugar, se realiza la presegmentación GMM, que además de modelos pertenecientes a segmentos “no-voz” (silencio, música, ruido...) puede incluir distintos modelos para la identificación de segmentos de voz (voz limpia, voz con música...). A continuación, se extraen los i-vectors correspondientes a los segmentos de voz y se realiza la clasificación de los mismos mediante

alguno de los métodos más frecuentemente utilizados (distancia coseno, distancia euclídea, SVM....). Por último, las etiquetas de voz obtenidas de la clasificación mediante i-vectors se proporcionan junto con las etiquetas no-voz obtenidas a partir de la segmentación de Viterbi.

3.3.2.1 **Validación del método propuesto**

La validación de la nueva técnica de segmentación de audio mediante i-vectors se ha llevado a cabo en la campaña de evaluación Albayzin 2014, que consistió de nuevo en la segmentación de audio broadcast, indicando qué segmentos contienen voz, música y ruido, pudiendo existir solapamiento entre las clases en cualquier instante. A diferencia de las ediciones anteriores, el objetivo de esta evaluación fue proporcionar un marco experimental mediante la combinación de distintas bases de datos, aumentando la dificultad de las últimas ediciones y poniendo a prueba la robustez de los sistemas participantes frente a diferentes contextos acústicos.

La base de datos utilizada, recogida en el capítulo 2, surge como combinación de tres bases de datos diferentes. Como se ha comentado anteriormente, los segmentos de audio que contienen voz representan más del 92 % del audio disponible en esta base de datos, mientras que dos clases minoritarias, ruido aislado y música con ruido, cuentan con una presencia inferior al 0.5 % y 0.3 % respectivamente, por lo que la técnica propuesta se ajusta perfectamente al problema propuesto por la organización de la evaluación.

Del total de grabaciones de la base de datos, 20 sesiones se destinaron al entrenamiento de los distintos sistemas, mientras que las 15 restantes fueron reservadas por la organización para la realización del test. Las 5 últimas sesiones de la parte de entrenamiento (15-20) se utilizaron como set de desarrollo para llevar a cabo la optimización de los parámetros del sistema desarrollado.

La figura 3.16 muestra el diagrama del sistema de segmentación de audio desarrollado con motivo de la celebración de la campaña de evaluación Albayzin 2014, y que hace uso de la técnica de propuesta [141]. La salida final del sistema de segmentación de audio propuesto se obtiene mediante la fusión de las etiquetas proporcionadas por dos subsistemas diferentes:

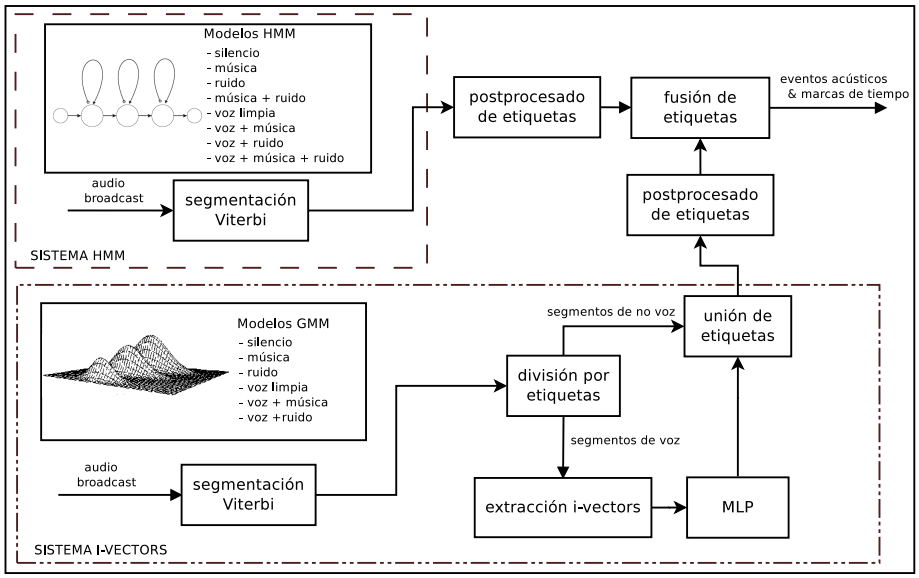


Figura 3.16: Diagrama del sistema de segmentación de audio Albayzin 2014

- Un sistema HMM mediante algoritmo de Viterbi, con 8 modelos diferentes para cada una de las clases no solapadas (silencio, voz, música, ruido, voz con música, voz con ruido, música con ruido y voz con música y ruido).

- Un enfoque diferente, basado en una presegmentación GMM y un refinamiento de los segmentos de voz mediante la clasificación de i-vectors llevada a cabo por un perceptrón multicapa.

La combinación de las salidas de los dos subsistemas se estudia en detalle en el capítulo 4, dedicado a la fusión. En este punto nos centraremos en los resultados obtenidos por los subsistemas individualmente, con especial énfasis en el sistema basado en la técnica propuesta de refinamiento mediante i-vectors.

El sistema basado en HMMs (con diseño y rendimiento similar al sistema propuesto en la campaña de 2012), establecerá el marco de referencia para el análisis del método propuesto. Utilizando las señales de entrenamiento de la base de datos, se entrena un modelo HMM independiente con 3-estados y mezclas de 512 gaussianas para las 8 clases definidas con anterioridad (silencio, voz limpia, música...).

Estos ocho modelos se utilizan en una segmentación mediante algoritmo de Viterbi para detectar las fronteras de los segmentos de audio que contienen los

Tabla 3.7: Error obtenido en términos de SER por el sistema basado en HMMs en las sesiones de desarrollo en función del número de gaussianas en las mezclas

Número de gaussianas	Segmentation Error Rate (SER)
32	28.28 %
64	28.19 %
256	27.79 %
512	**26.50 %**
1024	26.95 %

diferentes eventos acústicos. Para realizar la clasificación se utiliza una parametrización construida mediante 13 coeficientes MFCC añadiendo primeras y segundas diferencias. Para entrenar los modelos y llevar a cabo la segmentación de audio se ha utilizado el toolkit HTK [157].

Los experimentos realizados sobre las sesiones de desarrollo mostraron que el uso de mezclas de 512 gaussianas proporciona el error de segmentación más bajo. Se puede observar el resultado de dichos experimentos en la tabla 3.7.

A continuación, se describe en detalle el sistema basado en i-vectors. En primer lugar, 6 modelos GMM con mezclas de 32 gaussianas entrenados para las clases de silencio, música, ruido, voz limpia, voz con ruido, y voz con música, se utilizan en una segmentación mediante algoritmo de Viterbi para identificar los segmentos de silencio, música y en especial de voz, independientemente de la presencia de ruido o música (o ambos) de fondo. Se utilizan en este caso 12 coeficientes MFCC con primeras y segundas derivadas para llevar a cabo la clasificación del audio (no se incluye en la parametrización el coeficiente relacionado con la energía).

Una vez identificados los segmentos de voz, se lleva a cabo el proceso de extracción de los i-vectors, utilizando para ello un mecanismo de ventana deslizante. La longitud de la ventana se fija en 5 segundos con el fin de obtener un i-vector suficientemente representativo, mientras que se utiliza un desplazamiento de 1 segundo en base a la resolución aplicada en la campaña de evaluación a las fronteras entre etiquetas. A continuación, un MLP se encarga de clasificar cada i-vector como voz limpia, voz con ruido, voz con música o voz con música y ruido. Para entrenar

Tabla 3.8: Rendimiento del MLP en función de la dimensión de los i-vectors utilizados, extraidos de segmentos de audio las sesiones de entrenamiento de la base datos

Dimensión	Accuracy	Precisión	Recall	Fscore
150	79.91 %	0.7887	0.7901	0.7894
125	79.83 %	0.7886	0.7897	0.7891
100	**80.61 %**	**0.7985**	**0.8009**	**0.7997**
75	80.27 %	0.7935	0.7949	0.7942
50	79.18 %	0.7816	0.7830	0.7823
25	74.23 %	0.7278	0.7328	0.7303

el modelo MLP, se utilizaron todos los segmentos correspondientes a la parte de entrenamiento de la base de datos.

Un aspecto importante a tener en cuenta es la dimensión del i-vector adecuada, por lo que se construyeron, mediante el software WEKA [54], diferentes clasificadores MLP con distintos tamaños de vector y se utilizaron los segmentos de entrenamiento de la base datos en una validación cruzada de 10 capas. La tabla 3.8 muestra los resultados obtenidos en base a métricas diferentes, como son el porcentaje de decisiones correctas (Accuracy), Precisión, Recall y Fscore. Estas métricas, orientadas a la evaluación de la clasificación (en menor medida a la evaluación de la segmentación de audio), se describen en detalle en el capítulo 4. Todas indican que 100 es la dimensión adecuada para la clasificación de los segmentos de voz.

Ambos sistemas incluyen un bloque de postprocesado de etiquetas cuyo objetivo es refinar los límites de los segmentos, eliminar silencios demasiado cortos y corregir algunos errores menores cometidos en el proceso de segmentación. También se unifican los segmentos consecutivos con la misma etiqueta para mejorar el rendimiento de la etapa de fusión de etiquetas posterior.

Al igual que en las evaluaciones anteriores, la métrica utilizada para evaluar el funcionamiento de los sistemas participantes en la campaña Albayzin 2014 ha sido el SER o Error de Segmentación. La tabla 3.9 presenta los resultados obtenidos por ambos sistemas en las sesiones de entrenamiento y desarrollo de la base de datos. De nuevo, los resultados han sido obtenidos con las herramientas de evaluación proporcionadas por la organización de la campaña de evaluación.

Tabla 3.9: Resultados obtenidos por los dos subsistemas en las sesiones de entrenamiento y de desarrollo de la base de datos Albayzin 2014 en términos de SER

Sesión	HMM	i-vec	Sesión	HMM	i-vec
01	17.27 %	9.55 %	09	19.3 %	10.35 %
02	20.82 %	10.84 %	10	17.24 %	12.55 %
03	16.07 %	10.29 %	11	20.77 %	10.36 %
04	21.43 %	10.33 %	12	17.02 %	8.11 %
05	17.88 %	9.39 %	13	17.48 %	10.33 %
06	22.94 %	14.6 %	14	19.38 %	12.94 %
07	28.18 %	10.27 %	15	19.75 %	8.87 %
08	13.97 %	10.92 %			
1-15	19.38 %	**10.67 %**			
16	18.79 %	15.72 %	19	19.68 %	15.66 %
17	14.17 %	12.87 %	20	33.09 %	21.73 %
18	25.74 %	16.35 %			
16-20	21.99 %	**16.33 %**			

Como se muestra en la tabla 3.9, el sistema basado en i-vectors tiene un comportamiento ampliamente superior al mostrado por el sistema basado en HMMs con un 16.33 % y un 21.99 % de SER respectivamente en las sesiones de la parte de desarrollo de la base de datos. También se muestra una menor variabilidad en los valores obtenidos por el sistema basado en i-vectors, tanto en las sesiones de entrenamiento como en las de desarrollo, debido posiblemente a su mayor discriminación en las clases en las que la voz está presente, lo que demuestra ampliamente la validez de la técnica de mejora de la segmentación propuesta.

En las tablas 3.10 y 3.11, los distintos eventos acústicos (voz, música y ruido) son evaluados de forma individual. De esta forma podemos observar en detalle el SER obtenido por cada uno de los dos subsistemas.

La tabla 3.10 muestra el error de segmentación obtenido por el sistema basado en HMMs en las sesiones de desarrollo de la base de datos. Al igual que en ediciones anteriores, la principal fuente del error de segmentación reside en la detección de música y en mayor medida de ruido, con un Error de Clase obtenido

Tabla 3.10: Detalle del error cometido, en términos de SER, por el sistema basado en HMMs para las clases de voz, música y ruido de forma individual en las sesiones de desarrollo de la base de datos Albayzin 2014

Tipo de error	Voz	Música	Ruido
Error de Omisión	2.0 %	23.2 %	28.9 %
Error de Inserción	3.3 %	9.4 %	41.2 %
Error de Clase	5.3 %	32.6 %	70.1 %

del 32.6 % y del 70.11 % respectivamente, mientras que la clase de voz obtiene resultados aceptables con un 5.3 % de Error de Clase.

La tabla 3.11 muestra los resultados al utilizar el sistema basado en i-vectors. Al igual que en el caso del sistema basado en HMMs, la principal fuente del error de segmentación reside en la detección del ruido y música, aunque en este caso se consigue una mejora considerable en el rendimiento en comparación con el sistema basado en HMMs para ambas clases, con un 18.4 % de Error de Clase para la clase de música y un 46.5 % para el ruido. El etiquetado de la voz obtiene resultados ligeramente inferiores, con un Error de Clase de 7.6 %, debido al mayor Error de Omisión obtenido como resultado de la presegmentación voz/no voz menos precisa llevada a cabo por el clasificador GMM.

La reducción de error de segmentación en las clases de música y ruido no es consecuencia de una mejor clasificación de los eventos acústicos “sólo música ” y “sólo ruido”, sino de una mejor discriminación de los segmentos de “voz con música”, “voz con ruido” y ”voz con música y ruido“ llevada a cabo por el MLP.

Tabla 3.11: Detalle del error cometido, en términos de SER, por el sistema basado en i-vectors para las clases de voz, música y ruido de forma individual en las sesiones de desarrollo de la base de datos Albayzin 2014

Tipo de error	Voz	Música	Ruido
Error de Omisión	4.4 %	11.9 %	18.1 %
Error de Inserción	3.2 %	6.5 %	28.4 %
Error de Clase	7.6 %	18.4 %	46.5 %

Tabla 3.12: Tiempo de CPU invertido por ambos subsistemas (HMM e i-vectors) en procesar las sesiones de entrenamiento de la base de datos Albayzin 2014

Base de datos	Sistema HMM	Sistema i-vec	Fusión
15h 37m 50s	1h 39m 17s	8h 9m 17s	5s

La tabla 3.12 presenta el tiempo medio de CPU requerido para procesar las sesiones de entrenamiento de la base de datos. Nuevamente, los valores se mantienen por debajo del tiempo real en ambos sistemas (en quad-core 2.27 GHz, 32 GB RAM), siendo el sistema basado el i-vectors el más costoso computacionalmente.

Un análisis posterior del sistema reveló deficiencias en la optimización de recursos de la implementación del sistema i-vector. La gestión de archivos requerida entre la parametrización del audio y la extracción del i-vector acumula el 80 % del tiempo de procesado utilizado. La integración de estos dos pasos disminuiría considerablemente el tiempo de CPU mostrado en la Tabla 3.12 para dicho sistema.

Los resultados obtenidos en las sesiones de test de la base de datos quedan recogidos en la tabla 3.13. Se incluye además el error de segmentación obtenido al aplicar la etapa de fusión de etiquetas a las salidas proporcionadas por los dos sistemas y que refleja el resultado final del sistema propuesto en la campaña de evaluación Albayzin 2014. De nuevo el sistema basado en i-vectors tiene un rendimiento claramente superior al mostrado por el sistema basado en HMMs con un 22.47 % y un 27.37 % de SER respectivamente, lo que demuestra la validez de la técnica de segmentación propuesta. Ambos sistemas muestran sin embargo un aumento significativo en el error de segmentación respecto a la parte de desarrollo de

Tabla 3.13: Resultados obtenidos por el sistema de segmentación desarrollado en las sesiones de test de la base de datos Albayzin 2014 en términos de SER

Sistema	SER
HMM	27.37 %
i-vec	22.47 %
Fusión	20.68 %

Tabla 3.14: Detalle del error obtenido por el sistema desarrollado, en términos de SER, para las clases de voz, música y ruido de forma individual en las sesiones de test de la base de datos Albayzin 2014

Tipo de error	Voz	Música	Ruido
Error de Omisión	3.9 %	20.2 %	25.0 %
Error de Inserción	2.2 %	12.0 %	30.4 %
Error de Clase	6.1 %	32.2 %	55.4 %

la base de datos, lo que evidencia problemas de adaptación a nuevos datos. En este punto podría resultar interesante un bloque de mejora de la robustez del sistema frente a posibles variaciones en las condiciones del audio.

La tabla 3.14 muestra el error de segmentación obtenido por el sistema final (después de la fusión) para las clases de voz, música y ruido individualmente. Se puede observar cómo, al igual que en los casos anteriores, la principal fuente de error reside en las clases de música y ruido, con un Error de Clase del 32.2 % y del 55.4 % respectivamente, siendo el caso de la música el que presenta mayor aumento del error respecto a las señales de desarrollo. La clase de voz presenta igualmente tasas de error ligeramente superiores con 6.1 % de Error de Clase. A pesar de la dificultad añadida por la complejidad de la base de datos, el sistema propuesto se confirma como alternativa para la extracción de voz en audio broadcast.

En esta ocasión, 7 sistemas desarrollados por 4 grupos de investigación diferentes tomaron parte en la campaña de evaluación de segmentación de audio, siendo nuevamente el sistema propuesto el que obtuvo mejor resultado.

3.4 Conclusiones

En este capítulo se han descrito los distintos procesos y métodos involucrados en la tarea de segmentación de audio, implementada habitualmente mediante el modelado y la clasificación de muestras de audio de distintas clases acústicas.

En primer lugar, se ha llevado a cabo un exhaustivo análisis de la literatura en busca de diferentes técnicas de parametrización, modelado y clasificación del audio, con especial énfasis en los métodos más ampliamente utilizados: GMM, HMM SVM y ANN. Adicionalmente, se ha presentado la métrica más ampliamente utilizada en la literatura para realizar la evaluación del funcionamiento de los distintos sistemas de segmentación de audio desarrollados, el SER.

A continuación, se han presentado las dos técnicas propuestas para la mejora de la segmentación de audio, basadas en el postprocesado de segmentos de voz en busca de música de fondo de bajo nivel y en una segmentación más robusta mediante clasificación de i-vectors respectivamente.

Por último, con el fin de comprobar el buen funcionamiento las dos técnicas propuestas se han recogido los resultados de dos sistemas de segmentación desarrollados con motivo de la participación en las campañas de evaluación Albayzin 2012 y 2014, que implementan cada una de las técnicas propuestas.

En primer lugar se han presentado los resultados obtenidos por el sistema de segmentación de audio diseñado para la campaña de evaluación Albayzin 2012, que implementa la técnica basada en el postprocesado de segmentos de voz en busca de música de fondo de bajo nivel. En este caso el sistema diseñado ha demostrado el buen rendimiento de la técnica de postprocesado propuesta al obtener el mejor resultado de la evaluación en la que tomaron parte 6 sistemas desarrollados por 5 grupos de investigación diferentes.

Del mismo modo se han presentado los resultados obtenidos por el sistema de segmentación de audio diseñado para la campaña de evaluación Albayzin 2014, basado en la técnica de segmentación robusta mediante clasificación de i-vectors. De nuevo en este caso el sistema desarrollado ha demostrado el buen funcionamiento de la técnica propuesta al obtener el mejor resultado de la evaluación en la que tomaron parte 7 sistemas desarrollados por 4 grupos de investigación diferentes.

Yo hago lo que usted no puede y usted hace lo que yo no puedo. Juntos podemos hacer grandes cosas.

Madre Teresa de Calcuta

CAPÍTULO

4

Fusión

Al enfrentarse a problemas de clasificación como la segmentación de audio o el reconocimiento de locutores, es habitual desarrollar y comparar diferentes métodos, seleccionando finalmente el que obtiene los mejores resultados como sistema final. Sin embargo, si los errores cometidos por los diferentes métodos están suficientemente incorrelados, se podrían utilizar sus resultados para mejorar el rendimiento general del sistema seleccionado mediante diversas técnicas de fusión de clasificadores [77], [153]. En múltiples campañas de evaluación, por ejemplo, aun con objetivos de clasificación muy distintos, es frecuente llevar a cabo la fusión de varios sistemas y obtener así resultados superiores a los que proporciona cada uno de ellos individualmente [127].

En términos generales, hablaremos de fusión a tres niveles diferentes [120]:

- *a nivel de datos*: la fusión se realiza a partir de los datos provenientes de diversas fuentes, como sucede en la identificación de personas, combinando múltiples rasgos biométricos (voz, huella dactilar, imagen facial,...) [68].

- *a nivel de características*: la fusión se realiza a partir de distintos tipos de características extraídas de los datos de que se dispone para realizar la clasificación, como sucede en los sistemas de verificación de locutor, que utilizan información segmental y prosódica a partir de la voz de los locutores [114].

- *a nivel de clasificadores*: la fusión se realiza a partir de los resultados de los clasificadores directamente, como sucede en [152].

Hablaremos de fusión de clasificadores en este último caso. Adicionalmente, los métodos de fusión que trabajan directamente con la salida de los clasificadores se dividen tradicionalmente en tres tipos, según la clase de salida proporcionada por los clasificadores originales involucrados en el proceso:

- *tipo 1*: el único dato proporcionado a la salida es la clase asignada por cada clasificador original [111]. No se dispone de ninguna información acerca de la confianza del clasificador al emitir dicha clase.

- *tipo 2*: cada uno de los clasificadores proporciona a la salida una lista con las distintas clases ordenadas en función de la confianza obtenida para cada una de ellas [160].

- *tipo 3*: cada clasificador proporciona a la salida, además de la clase asignada, el nivel de confianza o *score* obtenida para cada una de las clases [22].

Los problemas de tipo 1, a nivel de etiqueta, donde los clasificadores proporcionan exclusivamente una etiqueta de clase a cada muestra de entrada, representan el escenario más complejo para la fusión, ya que la información proporcionada por los clasificadores es mínima (la menor posible). Sin embargo, también aparecen como caso más general, dado que cualquier tipo de clasificador desarrollado es capaz de proporcionar al menos etiquetas a su salida. Es por ello que la fusión de etiquetas puede suponer un reto interesante.

La fusión de etiquetas puede aplicarse a diversos problemas del mundo real, como puede ser la unificación de la información proporcionada por anotadores humanos expertos [111] o el desarrollo de sistemas colaborativos entre distintos laboratorios, ya que no requiere del intercambio de detalles específicos de los distintos clasificadores desarrollados individualmente. Otra ventaja que presenta la fusión de etiquetas es que permite una gran variabilidad en el tipo de clasificadores originales, pudiendo éstos ser de naturaleza muy diversa (como pueden ser clasificadores de tipo sintáctico [2]), ya que como se ha comentado anteriormente cualquier tipo de clasificador es capaz de proporcionar información sobre la clase a la que pertenece la muestra analizada.

Uno de los objetivos que se planteaban al comienzo de esta tesis era el desarrollo de técnicas que permitan aprovechar el conocimiento previo de los participantes de una determinada reunión de trabajo. La fusión de etiquetas puede resultar una buena alternativa en este caso proporcionando la salida combinada de varios sistemas de reconocimiento independientes. Adicionalmente, mediante el indexado previo de las salidas de los clasificadores originales, este tipo de fusión es capaz de proporcionar la salida del sistema de forma inmediata, permitiendo la toma de decisiones sobre la marcha, necesaria para extender el funcionamiento online a los sistemas de diarización.

Diversos métodos han sido propuestos para llevar a cabo una correcta fusión de etiquetas [81], [153], [78], [64]. Algunos de ellos tratan de sacar partido de la interacción entre los clasificadores originales, otros simplemente buscan otorgar un nivel de confianza a cada posible clase de salida. Estos últimos requieren a su vez de independencia estadística entre los clasificadores implicados en el proceso de fusión, aunque en la práctica esta condición no resulta indispensable para obtener un buen resultado [80], [3]. Todos ellos sin embargo, han sido construidos para mejorar el acierto o Accuracy (clasificar correctamente el mayor número de muestras posible) [49], por lo que tienden a realizar clasificaciones muy pobres cuando las bases de datos utilizadas presentan un gran desequilibrio entre sus clases.

Muchos de los problemas del mundo real presentan problemas de desequilibrio entre los datos, con algunas de las clases muy pobremente representadas con respecto a la clase o clases mayoritarias que concentran la mayoría de las muestras. Este es el caso, por ejemplo, del diagnóstico médico, la detección de intrusos, el reconocimiento de emociones o la clasificación de textos. Es por esto, que el problema del desequilibrio entre clases está considerado aún hoy en día uno de los grandes retos de la minería de datos [154].

La mayor parte de algoritmos de clasificación suponen equilibrio entre las clases presentes en la base de datos [57], algo que, en su esfuerzo por mejorar el número de aciertos, les lleva a asignar asiduamente las muestras analizadas a las clases más representadas. De esta forma se produce un descenso del rendimiento de los distintos métodos que afecta especialmente a la identificación de las muestras de las clases minoritarias. Este descenso se verá afectado en mayor o menor

medida dependiendo del grado de desequilibrio entre clases, la complejidad de los datos, el tamaño de la base de datos o del propio método de clasificación [71].

Las clases minoritarias son además las más interesantes (generalmente) desde el punto de vista del reconocimiento y aprendizaje, y la pérdida de una muestra de una de estas clases resulta a menudo más crítica que en el caso de las clases mayoritarias [84]. Por ello, diversos esfuerzos [16] [59] [47] se han llevado a cabo a la hora de construir clasificadores que respondan de forma más adecuada al problema de desequilibrio en los datos. Algunos hacen uso del remuestreo de los datos para equilibrar las distintas clases [24] mientras que otros aplican métodos de aprendizaje costo-sensitivo [39].

La fusión de etiquetas puede suponer una alternativa más sencilla para mejorar el funcionamiento de los clasificadores en este caso. Sin embargo, como se ha comentado anteriormente, ninguno de los métodos propuestos es capaz de lidiar de forma adecuada con el problema de desequilibrio entre clases. En este capítulo, se presenta una nueva técnica de fusión de etiquetas que tiene en cuenta el mencionado desequilibrio entre clases de la base de datos. Se han realizado además diversos experimentos para comparar el funcionamiento del algoritmo propuesto con otros métodos del estado del arte.

El resto del capítulo se organiza de la siguiente forma. En primer lugar, se presentan el problema de clasificación de patrones y los distintos métodos de fusión de etiqueta utilizados habitualmente en la literatura. A continuación, se examina el problema de la evaluación del rendimiento de los distintos métodos en entornos con desequilibrio en los datos. Posteriormente, se describe en detalle el método propuesto y se analiza el comportamiento de los distintos métodos enfrentados a un caso concreto de clasificación de datos desequilibrados. Se muestran además los resultados experimentales sobre múltiples bases de datos desequilibradas. Por último, se presenta una extensión del método de fusión propuesto para extender su funcionamiento a un nivel de score y se recogen los distintos experimentos realizados en el área del reconocimiento y verificación de locutores.

4.1 Estado del arte

4.1.1 Nomenclatura

Consideremos un problema de clasificación con un conjunto de etiquetas mutuamente excluyentes $\Lambda = \{1, 2, \cdots, M\}$ que representan un conjunto de patrones o clases y C un conjunto de K clasificadores con $C = \{c_1, \cdots, c_K\}$. Denotamos como $c_k(x) = l_k$ la tarea del clasificador c_k de asignar como salida la etiqueta l_k a una muestra de entrada x. De acuerdo al trabajo realizado en esta tesis no se contempla el posible rechazo de una muestra (el clasificador debe asignar una de las etiquetas disponibles a cada una de las muestras de entrada), por lo que al asignar la etiqueta l_k, vamos a considerar que la muestra x pertenece a la clase m_k, con $m_k \in \Lambda$.

4.1.2 Fusión de clasificadores y matriz de confusión

El objetivo de los diferentes métodos de fusión es crear un sistema conjunto E, a partir de los eventos producidos por los K clasificadores individuales, que asigne en cada caso la etiqueta óptima l_{opt} a la muestra de entrada x, es decir, que produzca el evento $E(x) = l_{opt}$.

La matriz de confusión $\mathbf{\Omega}^{(k)}$ de un clasificador es una forma habitual de caracterizar su comportamiento en el pasado, que ofrece información útil a la hora de construir el sistema conjunto, E, si se asume que el comportamiento previo del clasificador es de alguna forma representativo de su comportamiento en el futuro [106]. Dada una base de datos de entrenamiento, la matriz de confusión de un clasificador k puede ser calculada como:

$$\mathbf{\Omega}^{(k)} = \begin{pmatrix} n_{11}^{(k)} & n_{12}^{(k)} & \dots & n_{1M}^{(k)} \\ n_{21}^{(k)} & n_{22}^{(k)} & \dots & n_{2M}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ n_{M1}^{(k)} & n_{M2}^{(k)} & \dots & n_{MM}^{(k)} \end{pmatrix} \tag{4.1}$$

donde las filas se corresponden con las diferentes clases presentes en la base de datos y las columnas con las decisiones realizadas por el clasificador para cada clase. Cada elemento $n_{ij}^{(k)}$ representa el número de muestras de la clase i etiquetadas

como j por el clasificador c_k, de modo que los elementos de la diagonal representan las muestras clasificadas correctamente y los elementos fuera de la diagonal representan los errores cometidos por el clasificador.

La información almacenada en la matriz $\mathbf{\Omega}^{(k)}$ puede ser utilizada para estimar diferentes probabilidades que permitan cuantificar la confianza depositada en cada decisión del clasificador, así como calcular diferentes medidas objetivas del rendimiento del mismo en una base de datos concreta.

4.1.3 Algoritmos de fusión a nivel de etiqueta

Una vez establecida la nomenclatura y presentado el problema de la fusión de clasificadores vamos a ocuparnos los principales métodos de fusión a nivel de etiqueta que podemos encontrar en la literatura: votación por mayoría (Majority Voting), integración de creencias de Bayes (Bayes Belief Integration), matriz de fusión por pares (Pairwise Fusion Matrix) y espacio de conocimiento del comportamiento (Behaviour-Knowledge Space).

4.1.3.1 Majority Voting

La estrategia más habitual a la hora de combinar la decisión de varios clasificadores es sin duda la democracia, más conocida como Majority Voting en alguna de sus posibles variantes [81] [120]. Este algoritmo asigna a la muestra de entrada la etiqueta que recibe mayor número de votos entre los clasificadores involucrados en la fusión. Si definimos

$$T_k(x \in m) = \begin{cases} 1, \text{ si } l_k = m \text{ y } m \in \Lambda \\ \\ 0, \text{ en cualquier otro caso} \end{cases} \tag{4.2}$$

y

$$T_E(x \in m) = \sum_{k=1}^{K} T_k(x \in m), \quad m \in \Lambda \tag{4.3}$$

una regla simple para realizar un majority voting es la siguiente:

$$E(x) = \arg\max_{m=1}^{M}(T_E(x \in m)) \tag{4.4}$$

Se trata de un método de fusión sencillo, que presenta principalmente dos problemas. En primer lugar, puede ocurrir que varias etiquetas obtengan el mismo número de votos para una misma muestra de entrada, por lo que en estos casos, es necesario recurrir a una nueva estrategia que resuelva los posibles empates. En segundo lugar, al aplicar la regla descrita para realizar el majority voting, T_E simplemente cuenta los votos recibidos por cada una de las clases, de forma que cada evento $c_k(x) = l_k$ es tratado equitativamente independientemente del número y el tipo de errores cometido por cada clasificador individual.

4.1.3.2 **Bayes Belief Integration**

El método de Bayes Belief Integration (BBI) [153], establece que la clase con mayor valor de confianza debe ser elegida como salida del sistema final, por lo tanto, el valor de confianza en la predicción de cada una de las posibles clases debe ser calculado. Para cada clase m, con $m \in \Lambda$, este valor viene dado por:

$$bel(m) = \prod_{k=1}^{K} P(x \in m | c_k(x) = l_k) \tag{4.5}$$

Para poder aplicar esta ecuación, en teoría es necesario que exista independencia entre los clasificadores involucrados, sin embargo, trabajos en la literatura han demostrado que en la práctica el rendimiento no se ve afectado aunque esta condición no se mantenga [3]. La probabilidad de que la muestra de entrada x pertenezca a la clase m, dado que el clasificador c_k ha etiquetado la muestra como l_k, presente en la ecuación 4.5, puede ser estimada usando los valores almacenados en la matriz de confusión por medio de la expresión:

$$P(x \in m | c_k(x) = l_k) = \frac{n_{ml_k}^{(k)}}{\sum_{i=1}^{M} n_{il_k}^{(k)}}, \quad m \in \Lambda \tag{4.6}$$

Finalmente, se tiene en cuenta la hipótesis con mayor valor de confianza para asignar la salida final del sistema conjunto, esto es:

$$E(x) = \arg \max_{m=1}^{M} bel(m) \tag{4.7}$$

Como se observa en 4.7, se tienen en cuenta todas las posibles clases m para cada muestra de entrada x, lo que puede da lugar a errores a medida que el número

de clases M aumenta y crece la complejidad de la decisión de cada clasificador c_k. Otro problema importante a tener en cuenta, es que el algoritmo no tiene en cuenta la distribución de clases en la base de datos, por lo que si los datos no están equilibrados, el algoritmo tenderá a escoger como la salida la clase con mayor presencia en la misma.

4.1.3.3 **Pairwise Fusion Matrix**

El método Pairwise Fusion Matrix (PFM) trata de sacar partido de la interacción entre clasificadores [78]. Una PMF es una matriz tridimensional construida a partir las etiquetas reales de las muestras de una base de datos de entrenamiento y de las etiquetas de salida de dos clasificadores para dichas muestras. Dada una muestra de entrada x, para cada pareja de clasificadores c_k y $c_{k'}$ debemos calcular la probabilidad de que x pertenezca a la clase m, dado que el clasificador c_k ha etiquetado dicha muestra como l_k y el clasificador $c_{k'}$ como $l_{k'}$. Esta probabilidad puede ser estimada utilizando la información almacenada en la PFM, según:

$$P(x \in m | c_k(x) = l_k, c_{k'}(x) = l_{k'}) = \frac{n(m, l_k, l_{k'})}{n(l_k, l_{k'})} \tag{4.8}$$

donde $n(l_k, l_{k'})$ representa el número total de muestras en las que las etiquetas l_k y $l_{k'}$ han sido asignadas por los clasificadores c_k y $c_{k'}$, y $n(m, l_k, l_{k'})$ el número de muestras en $n(l_k, l_{k'})$ etiquetadas como m en la base de datos de entrenamiento.

Dada una muestra de entrada x etiquetada como l_k por el clasificador c_k y como $l_{k'}$ por el clasificador $c_{k'}$, la clase de salida más probable generada por esta pareja de clasificadores puede ser estimada como:

$$\hat{m}_{kk'} = \arg \max_{m=1}^{M} P(x \in m | c_k(x) = l_k, c_{k'}(x) = l_{k'}) \tag{4.9}$$

Esta clase de salida más probable debe ser estimada para cada una de las $K(K-1)/2$ parejas de clasificadores existentes. Finalmente, se toma como salida del sistema final la etiqueta que recibe mayor número de votos de dichas parejas, según:

$$T_{kk'}(x \in m) = \begin{cases} 1, \text{ si } \hat{m}_{kk'} = m \text{ y } m \in \Lambda \\ \\ 0, \text{ en cualquier otro caso} \end{cases} \tag{4.10}$$

$$T_E(x \in m) = \sum_{k,k'=1;k<k'}^{K} T_{kk'}(x \in m), \quad m \in \Lambda \tag{4.11}$$

$$E(x) = \arg\max_{m=1}^{M} T_E(x \in m) \tag{4.12}$$

El problema principal de este método es que se necesita un gran número de muestras en la base de datos para que la estimación de $P(x \in m|c_k(x) = l_k, c_{k'}(x) = l_{k'})$ pueda ser considerada representativa. Adicionalmente, para llevar a cabo la transformación PFM, es necesario calcular $K(K-1)/2$ matrices (una por cada pareja de clasificadores), lo que puede suponer un exceso de recursos a medida que el numero de clasificadores involucrados en el proceso de fusión aumenta.

4.1.3.4 **Behaviour-Knowledge Space**

Behaviour-Knowledge Space (BKS) [64] es un espacio de K dimensiones, donde cada dimensión se corresponde con las decisiones tomadas por cada uno de los clasificadores. Como ocurre con PFM, este método también está basado en la interacción entre los clasificadores involucrados en la fusión.

En primer lugar, en la etapa de modelado, cada posible combinación de las clases asignadas por los clasificadores individuales a las muestras de la base de datos de entrenamiento $(c_1(x) = l_1, c_2(x) = l_2, ..., c_K(x) = l_K)$ se utiliza como índice a una celda de una tabla de consulta llamada unidad BKS, $BKS(r)$ con $r = 1, .., M^{K+1}$. Cada celda almacena tres datos distintos: el número total de muestras encontradas con la correspondiente combinación de etiquetas $T_{BKS(r)}$, el número total de dichas muestras pertenecientes a cada clase real $n_{BKS(r)}(m)$ y la clase más representativa para esa unidad BKS, es decir, la clase real con mayor número de entradas en la unidad BKS.

A continuación, en la etapa de procesado, para una muestra de entrada desconocida x, las decisiones individuales de los clasificadores involucrados determinan la celda de la tabla BKS que será consultada para obtener la salida del sistema final. Esta celda se denomina *focus unit* o $BKS(FU)$. En base a la información

contenida en dicha celda, la salida se obtiene mediante la siguiente expresión:

$$E(x) = \begin{cases} & \text{si } T_{BKS(FU)}>0 \text{ y} \\ R_{BKS(FU)}, & \frac{\max\limits_{m=1}^{M}(n_{BKS(FU)}(m))}{T_{BKS(FU)}} \geq \theta \\ 0, & \text{en cualquier otro caso} \end{cases} \tag{4.13}$$

donde θ es la mínima proporción de la clase más representativa en la unidad BKS requerida para considerar dicha clase como la salida final. En las situaciones en las que $T_{BKS(FU)} = 0$ o $\frac{\max\limits_{m=1}^{M}(n_{BKS(FU)}(m))}{T_{BKS(FU)}} < \theta$, se sugiere la utilización de un método alternativo, como el Bayesiano por ejemplo, para conseguir una mejora de los resultados [65].

Al igual que ocurre con el PFM, este algoritmo necesita un gran número de muestras para conseguir que la estimación de la tabla BKS sea fiable, por lo que no se recomienda el uso de este método en problemas de clasificación con un elevado número de clases o demasiados clasificadores involucrados. De hecho, el requerimiento de memoria de la tabla BKS crece exponencialmente a medida que el número de clasificadores aumenta, lo que puede suponer una importante restricción para determinadas aplicaciones. Otro problema que presenta el algoritmo BKS es que sólo una de las posibles clases puede ser seleccionada como la más representativa en cada unidad BKS, por lo que los posibles empates deben ser tratados de manera arbitraria.

4.2 Medidas de evaluación en clasificación

En un problema de clasificación se definen en primer lugar el número de positivos verdaderos o "True Positives" (TP) como el número de casos clasificados correctamente como pertenecientes a la clase positiva, el número de falsos positivos o "False Positives" (FP) como el número de casos clasificados erróneamente como pertenecientes a la clase positiva, el número de negativos verdaderos o "True Negatives" (TN) como el número de casos correctamente descartados como pertenecientes a la clase positiva, y el número de falsos negativos o "False Negatives" (FN) como el número de casos descartados erróneamente como pertenecientes a la clase positiva cuando realmente corresponden a dicha clase.

A continuación, se dispone de múltiples medidas para evaluar el rendimiento de un clasificador. Una de las más sencillas y más ampliamente utilizada es la Accuracy, definida como la proporción de predicciones correctamente realizadas respecto al total de muestras en la base de datos, esto es:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.14}$$

Sin embargo, esta medida es completamente inadecuada en el caso de utilizar bases de datos desequilibradas, ya que otorga el mismo peso a todas las predicciones correctas independientemente del reparto de clases, lo que distorsiona la realidad del funcionamiento del clasificador [90].

Problemas de clasificación como la segmentación de audio, el reconocimiento de locutores o el reconocimiento de emociones presentan habitualmente bases de datos muy desequilibradas, con clases que disponen de gran cantidad de muestras frente a otras con una representación muy inferior, por lo que se precisa de medidas más adecuadas que permitan representar con mayor exactitud el comportamiento de los clasificadores en este tipo de aplicaciones.

En la literatura se recogen diversas medidas para tener en cuenta el desequilibrio de muestras entre clases [51], incluyendo Precisión, Recall, Fscore, Gmean, curvas ROC o el área bajo la curva ROC (AUC).

El valor de Precisión, definido como la relación entre el número de casos correctamente identificados y las hipótesis realizadas por el sistema, viene dado por:

$$P = \frac{TP}{TP + FP} \tag{4.15}$$

De forma paralela, el valor de Recall, también denominado tasa de positivos verdaderos o "True Positive rate" (TP_{rate}), y definido como la relación entre el número de casos correctamente identificados y el número total de muestras de la clase positiva en las marcas de referencia, viene dado por:

$$R = TP_{rate} = \frac{TP}{TP + FN} \tag{4.16}$$

A menudo resulta complicado resolver el compromiso entre Recall y Precision, por lo que para llevar a cabo una mejor comparación se utiliza generalmente el valor de Fscore, definido como la media armónica de dichos términos:

$$Fscore = \frac{2RP}{R + P} \tag{4.17}$$

Basada en la misma idea del compromiso entre Recall y Precision, se obtiene el valor de de Gmean como la media geométrica de estos dos valores, esto es:

$$Gmean = (RP)^{\frac{1}{2}} \tag{4.18}$$

La curva de Característica Operativa del Receptor o "Receiver Operating Characteristic" (ROC) es una representación gráfica de la tasa de positivos verdaderos frente a la tasa de falsos positivos para un clasificador binario en función del umbral de decisión. La tasa de falsos positivos o "False Positive rate" (FP_{rate}), definida como la relación entre el número de casos clasificados erróneamente como pertenecientes a la clase positiva y el número total de muestras de la clase positiva en las marcas de referencia, viene dada por:

$$FP_{rate} = \frac{FP}{FP + TN} \tag{4.19}$$

En el caso concreto de los clasificadores discretos, que entregan como única salida una serie de etiquetas, la evaluación del rendimiento del clasificador proporciona un único punto en el espacio ROC. En la figura 4.1 se muestra a modo

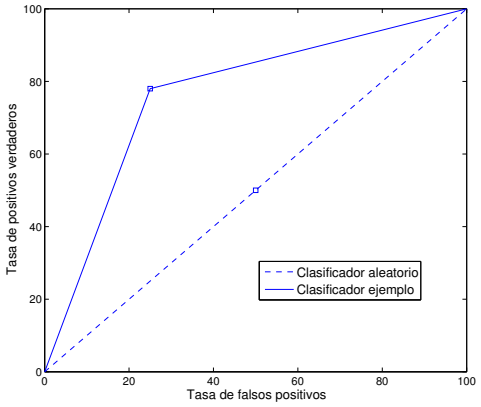


Figura 4.1: Ejemplo de curva ROC de un clasificador a nivel de etiqueta

de ejemplo la curva ROC de un clasificador cualquiera (además del clasificador aleatorio), que proporciona información a nivel de etiqueta exclusivamente.

Otro parámetro utilizado habitualmente para evaluar los resultados de un clasificador es el área bajo la curva ROC. Este área puede interpretarse como la probabilidad de que ante un par de muestras, una positiva y una negativa, el sistema las clasifique correctamente.En el caso de utilizar clasificadores que solo proporcionan información a nivel de etiquetas, el valor del AUC en un escenario bi-clase puede ser calculado como [49]:

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} \tag{4.20}$$

La mayor parte de las medidas descritas pertenecen al campo de la búsqueda y recuperación de información (information retrieval), donde por lo general existe un número elevado de documentos irrelevantes frente a los pocos relevantes disponibles, de forma que el problema puede ser reducido a una tarea de detección en un escenario bi-clase. En escenarios multi-clase como los que nos ocupan, el valor obtenido para cada clase debe ser combinado para proporcionar un valor único como medida final, ya sea calculando el valor medio por clase otorgando el mismo peso a todas las clases (macro-averaging), o calculando el valor medio total otorgando el mismo peso a todas las decisiones realizadas (micro-averaging) [135]. Al realizar micro-averaging se favorece la clase mayoritaria y al utilizar macro-averaging

se consideran todas las clases equitativamente, por lo que en caso de disponer de bases de datos desequilibradas el macro-averaging resulta más conveniente.

El valor de Recall una vez realizado el macro-averaging se denomina habitualmente Unweighted Average Recall (UAR), y ha sido utilizado ampliamente por múltiples autores como medida de evaluación en distintos campos como el reconocimiento de emociones a partir de voz [10], reconocimiento de la calidad de interacción de sistemas de diálogo [124], detección del autismo [126], detección de pitch [112], etc. Se trata de una medida con características interesantes para evaluar el rendimiento de un clasificador cuando los datos están desequilibrados [118]. Siguiendo la nomenclatura expuesta en la sección 4.1, se puede realizar el cálculo del UAR mediante la siguiente expresión:

$$UAR = \frac{1}{M} \sum_{i=1}^{M} \frac{n_{ii}^{(k)}}{\sum_{j=1}^{M} n_{ij}^{(k)}} \tag{4.21}$$

De forma similar, en diversos estudios recientes relativos a la evaluación del rendimiento de distintos algoritmos de clasificación en escenarios multi-clase con datos desequilibrados, se ha adoptado una versión extendida de la Gmean [122]. Se define como la media geométrica de los valores de recall de cada una de las clases presentes en la base de datos:

$$Gmean = \left(\prod_{i=1}^{M} \frac{n_{ii}^{(k)}}{\sum_{j=1}^{M} n_{ij}^{(k)}} \right)^{\frac{1}{M}} \tag{4.22}$$

Esta medida presenta la particularidad de penalizar enormemente un comportamiento especialmente negativo en alguna de las clases por parte del clasificador.

Adicionalmente, una versión generalizada del AUC denominada MAUC [56] está siendo ampliamente aceptada en la evaluación del rendimiento de clasificadores con datos desequilibrados. Se obtiene como el valor medio del AUC de cada posible pareja de clases, esto es:

$$MAUC = \frac{2}{M \cdot (M-1)} \sum_{i<j} \frac{AUC_{ij} + AUC_{ji}}{2} \tag{4.23}$$

donde AUC_{ij} (AUC_{ji}) es el valor del AUC de un clasificador en un problema binario que solo considera las muestras correspondientes a las clases i y j. Cabe

resaltar que en caso de estar tratando con un problema multi-clase AUC_{ij} y AUC_{ji} podrían presentar valores distintos.

En este trabajo hemos tomado los valores de UAR, Gmean y MAUC para evaluar el funcionamiento de los distintos algoritmos de fusión a nivel de etiqueta, ya que han sido utilizados ampliamente por múltiples autores en el campo de la clasificación en un entorno desequilibrado y multi-clase.

Adicionalmente, se proporcionan valores de Accuracy y Fscore en los experimentos destinados a determinar las diferencias entre los diferentes métodos en distintos escenarios. Además de la Accuracy, los autores a menudo recogen valores de Fscore por clase, por lo que se ha considerado que la inclusión de ambas medidas puede resultar de interés. En este caso, se aplica de nuevo el macro-averaging para mantener la coherencia con el resto de medidas utilizadas, proporcionando un valor único para evaluar el rendimiento global de cada algoritmo evaluado.

Los experimentos realizados en segmentación de audio y reconocimiento de locutores cuentan con métricas específicas de cada campo y serán introducidas junto con los propios experimentos.

4.3 Método de fusión propuesto

La técnica de fusión propuesta consiste en estimar la confianza en la decisión de un clasificador como la probabilidad de que la salida de dicho clasificador sea correcta y el resto de clasificadores hayan confundido la clase asignada a la muestra con la etiqueta propuesta por el clasificador cuya confianza se está evaluando.

Si los clasificadores que toman parte en el proceso de fusión han sido entrenados correctamente, es de suponer que al menos uno de ellos será capaz de asignar la etiqueta correcta a cada una de las muestras de entrada. Por lo tanto, a diferencia de lo que ocurre en el método de Bayes Belief Integration, donde la confianza en la decisión se evalúa para cada clase posible ($bel(m)$, $m = 1, 2, \cdots, M$), lo que se propone es evaluar dicha confianza para la decisión tomada por cada uno de los clasificadores involucrados en la fusión ($con(k)$, $k = 1, 2, \cdots, K$).

La estimación de la confianza en la decisión de cada clasificador se expresa como el producto de dos términos. El primero representa la precisión del clasificador para la clase emitida, mientras que el segundo representa la probabilidad de que el resto de clasificadores hayan confundido dicha clase con la asignada por cada uno de ellos a la muestra de entrada. Este enfoque implica, por tanto, el producto de probabilidades, por lo que el método propuesto requiere de la existencia de independencia entre clasificadores.

Si denotamos por l_r la etiqueta asignada por el clasificador c_r, podemos estimar el valor de confianza en la decisión del clasificador c_k, es decir, $con(k)$, mediante la siguiente expresión:

$$con(k) = P(x \in m_k | c_k(x) = l_k) \prod_{\substack{r=1 \\ r \neq k}}^{K} P(c_r(x) = l_r | x \in m_k) \tag{4.24}$$

donde $con(k)$ expresa el valor de confianza en la decisión del clasificador c_k, basado en su precisión para la clase m_k y en la probabilidad de que el resto de clasificadores $c_r, r \neq k$ hayan etiquetado erróneamente como l_r la clase m_k.

La probabilidad $P(x \in m_k | c_k(x) = l_k)$ es estimada por medio de (4.6) considerando simplemente $m = m_k$ para cada uno de los clasificadores según:

$$P(x \in m_k | c_k(x) = l_k) = \frac{n^{(k)}_{m_k l_k}}{\sum_{i=1}^{M} n^{(k)}_{i l_k}} \tag{4.25}$$

donde $n^{(k)}_{m_k l_k}$ denota el número de muestras de la clase m_k etiquetadas correctamente por el clasificador c_k como l_k. O visto de otra manera, este término representa el valor de precisión del clasificador c_k para la clase m_k.

Asumiendo que c_k ha asignado correctamente la etiqueta l_k a la muestra x, es decir, considerando que x pertenece a la clase m_k, cada elemento del segundo término en (4.24) viene dado por:

$$P(c_r(x) = l_r | x \in m_k) = \frac{n^{(r)}_{m_k l_r}}{\sum_{j=1}^{M} n^{(k)}_{m_k j}} \tag{4.26}$$

donde $n^{(r)}_{m_k l_r}$ representa el número de muestras de la clase m_k etiquetadas como l_r por el clasificador c_r en la parte de entrenamiento de la base de datos. En este caso, se utiliza el número total de muestras de la clase m_k en la base de datos para realizar la estimación de la probabilidad, por lo que puede ser calculado sumando los valores de la fila correspondiente en cualquiera de las matrices $\mathbf{\Omega}^{(k)}$.

La clase l_{opt} seleccionada por el clasificador c_{opt} cuyo valor de confianza es mayor, se toma como etiqueta final del sistema, según:

$$E(x) = l_{opt}, \text{ si } c_{opt} = \arg \max_{k=1}^{K} con(k) \tag{4.27}$$

Cabe resaltar que si un clasificador c_r no ha etiquetado en ningún caso la clase m_k como l_r, tenemos que $n^{(r)}_{m_k l_r}$ es 0. Según lo establecido en (4.24), esto produce un valor final de confianza $con(k)$ de 0, lo que puede originar que varios clasificadores acumulen exactamente el mismo valor de $con(k)$. Para evitar este efecto, los ceros de la matriz de confusión deben ser sustituidos por un valor mínimo, que puede ser ajustado empíricamente en la parte de entrenamiento de la base de datos.

Con la utilización del método propuesto se pretende refinar el etiquetado final del sistema, teniendo en cuenta la decisión por el clasificador cuyo valor de confianza es mayor. Así pues, se ha denominado al método propuesto como algoritmo TRURG (TaggeR Using most Reliable Guess).

4.4 Fusión de etiquetas vs datos desequilibrados

En esta sección se realiza el estudio del comportamiento de los distintos métodos de fusión descritos cuando se enfrentan a un problema de desequilibrio en la base de datos. Se utiliza para ello un caso extraído de los experimentos recogidos en la Sección 4.5, que será analizado en detalle en lo que al funcionamiento interno de los métodos de fusión se refiere, no al resultado obtenido en cada caso.

Consideremos por tanto un problema de clasificación con una base de datos con $M = 4$ etiquetas de clase y un conjunto de $K = 3$ clasificadores. La base de datos contiene 3520 muestras que siguen la siguiente distribución: 214 muestras de la clase 1, 203 muestras de la clase 2, 320 muestras de la clase 3 y 2783 muestras de la clase 4. Cabe resaltar en este punto que toda la información proporcionada en el ejemplo ha sido extraída de un experimento real (no se trata de valores escogidos al azar). Las siguientes matrices de confusión $\mathbf{\Omega}^{(1)}$, $\mathbf{\Omega}^{(2)}$ y $\mathbf{\Omega}^{(3)}$ han sido obtenidas para los clasificadores c_1, c_2 y c_3 respectivamente, utilizando para ello la parte de entrenamiento de la base de datos.

$$\mathbf{\Omega}^{(1)} = \begin{pmatrix} 129 & 22 & 18 & 45 \\ 10 & 97 & 37 & 59 \\ 31 & 36 & 158 & 95 \\ 207 & 227 & 356 & 1993 \end{pmatrix}$$

$$\mathbf{\Omega}^{(2)} = \begin{pmatrix} 95 & 26 & 19 & 74 \\ 21 & 63 & 51 & 68 \\ 23 & 39 & 164 & 94 \\ 365 & 446 & 572 & 1400 \end{pmatrix}$$

$$\mathbf{\Omega}^{(3)} = \begin{pmatrix} 141 & 8 & 10 & 55 \\ 11 & 113 & 26 & 53 \\ 13 & 34 & 164 & 109 \\ 148 & 143 & 256 & 2236 \end{pmatrix}$$

Supongamos ahora que $c_1(x) = l_1 = 3$, $c_2(x) = l_2 = 3$ y $c_3(x) = l_3 = 2$ han sido las etiquetas asignadas por los clasificadores originales c_1, c_2 y c_3 respectivamente a una nueva muestra de entrada concreta x.

Basando la decisión en el enfoque del Majority Voting, según queda reflejado al aplicar la eq. (4.4), tomaremos la clase 3 como etiqueta final del sistema, que es la que ha recibido mayor número de votos por parte de los clasificadores originales.

Nótese cómo los eventos $c_1(x)$, $c_2(x)$ y $c_3(x)$ han sido tratados de forma equitativa independientemente del comportamiento previo de los clasificadores en la parte de entrenamiento de la base de datos. La decisión puntual tomada por cada clasificador es la única información utilizada para seleccionar la etiqueta final del sistema, lo que puede llevar a un comportamiento irregular en algunos casos.

De acuerdo con el método BBI, en primer lugar debemos calcular los valores de confianza en la predicción de que la muestra x pertenezca a cada una de las posibles clases por medio de la eq. (4.5).

$$
\begin{aligned}
bel(1) &= 18/569 \cdot 19/806 \cdot 8/298 = 0 \\
bel(2) &= 37/569 \cdot 51/806 \cdot 113/298 = 0.002 \\
bel(3) &= 158/569 \cdot 164/806 \cdot 34/298 = 0.006 \\
bel(4) &= 356/569 \cdot 572/806 \cdot 143/298 = 0.213
\end{aligned}
$$

A continuación, como queda recogido en la eq. (4.7), la clase con mayor valor de confianza obtenido es seleccionada como salida final del sistema. En este ejemplo concreto será la clase 4. Se puede observar cómo todos los valores de confianza calculados aparecen normalizados por los mismos términos (todos ellos están basados únicamente en el comportamiento previo de cada uno de los clasificadores originales para la clase emitida), por lo que sólo el número de casos almacenados en las matrices de confusión se ha tenido en cuenta a la hora de tomar la decisión final. Entornos de alto grado de desequilibrio en los datos, como el descrito en el ejemplo, donde la mayor parte de las muestras pertenecen a la clase 4, conducen al algoritmo a asignar continuamente esta etiqueta a cada nueva muestra de entrada, incluyendo aquellos casos en los que ninguno de los clasificadores originales la ha seleccionado previamente. Desde el punto de vista estadístico la clase mayoritaria es la etiqueta más probable para cada una de las nuevas muestras de entrada y el valor de Accuracy obtenido refrendará dicha hipótesis. Sin embargo, la clasificación general realizada resultará deficiente, ya que ninguna de las muestras de las clases minoritarias será etiquetada correctamente.

Si nos basamos en el algoritmo PFM, la probabilidad de que la nueva muestra de entrada x pertenezca a cada una de las posibles clases debe ser estimada para cada par de clasificadores implicados en el proceso de fusión, como queda recogido en la eq. (4.9). De nuevo, únicamente el número de casos almacenados en cada PFM con $l_1 = 3$, $l_2 = 3$ y $l_3 = 2$ se tendrán en cuenta para llevar a cabo la decisión final,

ya que todos los valores de probabilidad estimados aparecen normalizados por los mismos términos. Supongamos los siguientes números de casos almacenados en las matrices PFM (recordemos que los valores forman parte de un experimento real).

Matriz PFM	Clase 1	Clase 2	Clase 3	Clase 4	Representativa
c_1 y c_2	4	21	104	178	4
c_1 y c_3	3	17	15	32	4
c_2 y c_3	0	27	18	42	4

Aplicando la eq. (4.12) tomaremos como salida final la etiqueta que ha recibido mayor número de votos, que de nuevo resulta ser la clase 4. Al igual que ocurre en el caso de la confianza Bayesiana, la probabilidad de que la nueva muestra pertenezca a cada una de las posibles clases depende exclusivamente del número de casos almacenados en las matrices PFM. Dado que el número de muestras de la clase 4 es significativamente mayor al número de muestras del resto de clases, la mayoría de combinaciones de los pares de clasificadores terminarán por seleccionar dicha clase como representativa del par. Más aún, si en algún caso la decisión tomada por uno de los pares resulta diferente a la clase mayoritaria, el proceso de voto posterior puede imponer dicha clase como etiqueta final del sistema. Determinadas muestras de las clases menos representadas pueden ser clasificadas correctamente en aquellos casos con cierto consenso entre los clasificadores originales.

Una observación similar puede ser realizada en torno al método BKS, donde la decisión final del sistema está basada exclusivamente en la información almacenada en la unidad BKS. Consideremos los siguientes números de casos almacenados en dicha unidad formada mediante las clases asignadas por los clasificadores individuales del ejemplo durante la etapa de entrenamiento (recordemos de nuevo que dichos datos forman parte de un experimento real).

Clase 1	Clase 2	Clase 3	Clase 4	Representativa
0	9	9	14	4

En este ejemplo encontramos 32 casos con la combinación $l_1 = 3$, $l_2 = 3$ y $l_3 = 2$ en la parte de entrenamiento de la base de datos, la mayoría de los cuales pertenecen a la clase 4. Esta etiqueta será designada como clase más representativa de la unidad BKS y seleccionada por tanto como salida final del sistema. De nuevo, el número de muestras de la clase 4 en la base de datos está condicionando la

decisión final del método. La combinación de decisiones involucra ahora a más de dos clasificadores (recordemos que PFM trabaja con pares de clasificadores), por lo que la selectividad aumenta y una mejor clasificación de las muestras de las clases minoritarias es posible en este caso cuando existe cierto consenso entre los clasificadores originales involucrados en la fusión. No obstante, la clasificación general seguirá lejos de ser óptima, ya que la clase mayoritaria será asignada asiduamente a las nuevas muestras en los casos en los que las etiquetas seleccionadas por los clasificadores originales sean diferentes.

Finalmente, si basamos la decisión en el método TRURG, debemos estimar en primer lugar los valores de confianza en la decisión tomada por cada uno de los clasificadores originales según la eq. (4.24)

$$\begin{aligned} con(c_1) &= 158/569 \cdot 164/320 \cdot 34/320 = 0.015 \\ con(c_2) &= 164/806 \cdot 158/320 \cdot 34/320 = 0.011 \\ con(c_3) &= 113/298 \cdot 37/203 \cdot 51/203 = 0.017 \end{aligned}$$

A continuación, siguiendo la eq. (4.27), la clase seleccionada por el clasificador que ha obtenido un valor de confianza más elevado, es asignada finalmente a la muestra de entrada. En este ejemplo concreto será la clase 2. Se puede observar cómo los valores de confianza estimados aparecen normalizados por términos diferentes en este caso. Por una parte, se estima la precisión de los clasificadores para la clase emitida, que es diferente para cada clasificador y cada clase. Por otro lado, asumiendo correcta la clase asignada por cada clasificador a la muestra, se aplica un factor de penalización basado en las decisiones de los dos clasificadores restantes, que no depende exclusivamente del número de casos almacenado en la matriz de confusión, sino también del número de muestras de la clase asignada por cada clasificador presentes en la base de datos. En otras palabras, no se tiene en cuenta el número de casos almacenado durante la etapa de entrenamiento (como ocurre al aplicar los métodos descritos anteriormente), sino cuán representativos resultan esos casos en la base de datos. Ciertamente, múltiples casos de la clase mayoritaria serán etiquetados de forma errónea, ya que un elevado número de muestras supone también un elevado factor de penalización. Sin embargo, se perderá un número escaso de muestras de las clases minoritarias y la clasificación general resultará significativamente más precisa.

4.5 Validación del método propuesto

Para validar el algoritmo propuesto en la Sección 4.3 se han realizado dos conjuntos de experimentos. En el primer grupo se ha utilizado una colección de bases de datos del repositorio UCI [83] junto con diferentes clasificadores entrenados mediante el software WEKA [54]. En el segundo grupo de experimentos realizados, el método propuesto ha sido aplicado a problemas propios del área del procesado de la voz, para lo que se han utilizado bases de datos y clasificadores específicos de la tecnología en este caso.

4.5.1 Condiciones de experimentación

En ambos conjuntos de experimentos, el método propuesto ha sido comparado con los diferentes algoritmos de fusión de etiquetas descritos en la sección 4.1.3. Para garantizar la repetitividad y reproducibilidad en los experimentos, se han realizado los siguientes ajustes en lo que a los distintos métodos de fusión se refiere.

- Majority voting (MAJ): en caso de producirse un empate en el número de votos recibidos por clases diferentes, se asignará a la salida la clase emitida por el clasificador con mayor Accuracy en la base de datos de entrenamiento.

- Bayes Belief Integration (BBI): cuando una confusión entre la clase i y la clase j no se ha dado en la base de datos de entrenamiento, n_{ij} toma valor cero, lo que lleva a muchos valores de confianza del mismo valor. Para evitar este efecto, se ha establecido un valor mínimo de 10^{-3} para cada elemento de la matriz de confusión.

- Pairwise Fusion Matrix (PFM): si más de una clase recibe el mismo número de votos para una pareja de clasificadores, la ecuación 4.9 produce varios posibles candidatos como clase representativa para este par. En este caso, la clase con índice más bajo se ha seleccionado como salida de dicha pareja de clasificadores. Adicionalmente, los empates producidos al obtener varias clases el mismo número de votos de todas las parejas de clasificadores, se han resuelto tomando la clase asignada por la pareja de clasificadores con más casos en la base de datos de entrenamiento.

- Behaviour Knowledge Space (BKS): Durante la etapa de modelado no es posible determinar la clase representativa en una unidad BKS si el número de casos para más de una clase coincide, por lo que la clase con índice más bajo entre los posibles candidatos se ha seleccionado como representativa de la unidad BKS en este caso. El método Bayes Belief Integration ha sido aplicado para obtener la decisión final en los casos que no han aparecido en la base de datos de entrenamiento.

Para realizar la comparación de los diferentes métodos se han utilizado las métricas descritas en la Sección 4.2: Accuracy, UAR, Gmean, MAUC y Fscore. Se ha prestado especial interés a los valores de UAR, Gmean y MAUC, utilizados ampliamente en el campo de la clasificación en un entorno desequilibrado y multiclase, como pueden ser la segmentación de audio, el reconocimiento de locutores o el reconocimiento de emociones.

4.5.2 Experimentos con las bases de datos UCI

Para llevar a cabo estos experimentos se han utilizado 18 bases de datos del repositorio UCI con distintas condiciones de desequilibrio entre clases, cuantificado mediante la tasa de desequilibrio o imbalance ratio (IR). Esta tasa se define como la relación entre el número de muestras de la clase mayoritaria y el número de muestras de la clase minoritaria dentro de una base de datos [101]. En general, una base de datos con IR menor a 1.5 se considera que está equilibrada [44].

La Tabla 4.1 recoge las principales características de las bases de datos UCI utilizadas en estos experimentos: el número total de muestras o instancias, el número de características o atributos que presenta cada muestra, el número de clases y la tasa de desequilibrio (IR) para cada una de las bases de datos. Como se puede observar, la tabla ha sido ordenada de acuerdo con este último parámetro en orden ascendente, es decir, de menor a mayor desequilibrio entre clases.

Cada base de datos de la Tabla 4.1 ha sido dividida aleatoriamente en dos partes: la parte de entrenamiento con el 50 % de las muestras, y la parte de test con el 50 % restante. La parte de entrenamiento ha sido dividida a su vez en dos sets mediante un nuevo proceso de selección aleatoria. Este proceso de división de la parte de entrenamiento de la base de datos se ha realizado sobre 100 repeticiones con el fin

Tabla 4.1: Características de las bases de datos UCI utilizadas en los experimentos

Base de datos	Instancias	Atributos	Clases	IR
Vowel	990	11	11	1.00
Segment	2310	20	7	1.00
Waveform	5000	41	3	1.02
Pendigits	10992	17	10	1.08
Vehicle	846	19	4	1.10
Letters	20000	17	26	1.10
Magic	19020	11	2	1.84
Pima	768	9	2	1.87
Thyroid	215	6	3	5.00
Dermatology	366	35	6	5.60
Balance	625	5	5	6.32
Glass	214	10	6	8.44
Soybean	683	36	19	11.50
Lymphography	148	19	4	40.50
Ecoli	336	8	8	71.50
Anneal	898	39	5	85.50
Yeast	1484	9	10	92.60
Hypothyroid	3772	30	4	1740.50

de obtener resultados confiables. En cada repetición, el 50 % de las muestras ha sido seleccionado de forma aleatoria para entrenar los clasificadores originales y el 50 % restante para realizar el cálculo de las matrices de confusión y entrenar los algoritmos de fusión de etiquetas analizados. Las medidas utilizadas para evaluar los distintos métodos de fusión (Accuracy, UAR, Gmean, MAUC y Fscore) han sido calculadas en cada repetición sobre la parte de test de la base de datos y los valores medios obtenidos se presentan como resultado final.

En primer lugar, se han utilizado las etiquetas proporcionadas por tres clasificadores entrenados por medio del software WEKA: un árbol de decisión rápida con poda de error reducida (fast decision tree learner with reduced error pruning) [41],

el clasificador de Bayes ingenuo con estimación de clases (Naive Bayes classifier using estimator classes) [72] y una sencilla tabla de decisión por mayoría (decision table majority classifier) [79], todos ellos con la configuración por defecto establecida en WEKA. La elección de estos clasificadores se ha realizado de forma arbitraria y otros podrían haber sido utilizados. Sin embargo, la mayoría de las bases de datos del repositorio UCI cuentan con un reducido número de muestras, clases y/o atributos, por lo que a menudo no suponen un reto para clasificadores más complejos, capaces de procesar correctamente la totalidad de las muestras en la fase de entrenamiento. En estos casos resulta imposible extraer información sobre los errores cometidos por los clasificadores individualmente, por lo que ninguna de las técnicas de fusión de etiquetas descritas resultaría efectiva en la fase de test.

Los resultados obtenidos se recogen en las tablas 4.2 y 4.3. Se muestra en negrita el mejor resultado en cada experimento y en cursiva los valores que no presentan diferencias estadísticamente significativas con éste. En ambos casos se utiliza además el color verde para facilitar la localización de dichos valores. Para llevar a cabo las comparaciones estadísticas se ha utilizado el test Wilcoxon con un nivel de significación del 95 %. Adicionalmente, se han incluido como referencia los resultados de un sistema Oracle (ORA), que elige la etiqueta correcta si cualquiera de los clasificadores originales ha sido capaz de seleccionarla previamente.

Las bases de datos han sido separadas en dos grupos en base al grado de desequilibrio entre clases. La Tabla 4.2 muestra el resultado obtenido al utilizar bases de datos equilibradas (IR$<$1.5). Se puede observar cómo BKS es el método que presenta un mejor funcionamiento en este caso en lo que a las medidas de Accuracy (5 de 6 casos), UAR (5 de 6 casos), Gmean (5 de 6 casos) y Fscore (5 de 6 casos) se refiere, aunque los tests de significación no muestran diferencias estadísticamente significativas en gran parte de los valores. Los resultados obtenidos referentes al MAUC son similares para todos los métodos analizados. Los métodos TRURG (definido como TRG en las Tablas), BBI y PFM han obtenido resultados similares con pequeñas diferencias en la mayoría de las bases de datos, mientras que MAJ se muestra como el método con peor comportamiento en este caso.

En este caso todas las medidas utilizadas resultan adecuadas para evaluar el funcionamiento de los distintos métodos, ya que cuando se utilizan bases de datos

Tabla 4.2: Resultados obtenidos al aplicar los métodos de fusión de etiquetas a los clasificadores "sencillos" utilizando bases de datos equilibradas (en negrita el mejor resultado en cada caso y en cursiva los valores que no presentan diferencias estadísticamente significativas con éste, ambos casos en verde)

		ORA	MAJ	BBI	PFM	BKS	TRG			ORA	MAJ	BBI	PFM	BKS	TRG
Vowel	Acc.	0.770	0.593	0.617	0.615	**0.626**	*0.618*	Segment	Acc.	0.975	0.926	*0.934*	*0.934*	**0.935**	*0.934*
	UAR	0.775	0.599	0.625	0.623	**0.633**	0.624		UAR	0.975	0.926	**0.935**	*0.934*	**0.935**	*0.934*
	Gmean	0.764	0.577	*0.586*	0.595	**0.608**	*0.607*		Gmean	0.974	0.922	**0.932**	*0.931*	**0.932**	*0.931*
	MAUC	0.934	**0.897**	*0.896*	0.890	0.884	**0.897**		MAUC	0.990	0.978	**0.979**	**0.979**	**0.979**	**0.979**
	Fscore	0.783	0.607	*0.627*	0.627	**0.635**	0.625		Fscore	0.975	0.927	*0.935*	*0.934*	**0.936**	*0.934*
Waveform	Acc.	0.938	0.791	0.796	0.798	**0.805**	0.795	Pendigits	Acc.	0.971	0.901	0.920	0.930	**0.933**	0.920
	UAR	0.938	0.791	0.796	0.798	**0.805**	0.795		UAR	0.970	0.900	0.920	0.929	**0.932**	0.919
	Gmean	0.936	0.784	0.790	0.787	**0.797**	0.790		Gmean	0.970	0.896	0.918	0.928	**0.931**	0.918
	MAUC	0.961	0.864	0.865	*0.876*	**0.877**	0.865		MAUC	0.992	0.976	0.981	**0.982**	**0.982**	0.981
	Fscore	0.940	0.794	0.797	0.802	**0.807**	0.797		Fscore	0.971	0.901	0.920	0.930	**0.932**	0.920
Vehicle	Acc.	0.811	*0.637*	0.630	*0.642*	0.630	**0.643**	Letters	Acc.	0.858	0.749	0.773	0.774	**0.781**	0.772
	UAR	0.804	*0.625*	0.622	**0.633**	0.623	*0.631*		UAR	0.857	0.749	0.772	0.773	**0.780**	0.771
	Gmean	0.776	0.537	0.454	0.525	*0.554*	**0.569**		Gmean	0.856	0.744	0.768	0.770	**0.777**	0.768
	MAUC	0.911	0.813	0.812	**0.822**	0.812	*0.817*		MAUC	0.956	0.931	**0.936**	0.935	0.926	**0.936**
	Fscore	0.810	*0.619*	0.606	**0.625**	*0.619*	*0.623*		Fscore	0.860	0.754	0.775	0.778	**0.783**	0.774

equilibradas, un valor elevado de Accuracy conlleva altos valores de UAR, Gmean, MAUC y Fscore, y por tanto, una mejor clasificación general de las muestras.

La Tabla 4.3 muestra el resultado obtenido al utilizar bases de datos con mayor grado de desequilibrio. En este caso los métodos PFM y BKS han conseguido mayores valores de Accuracy. Sin embargo, se puede observar cómo este hecho no garantiza una mejor clasificación general cuando existe desequilibrio en las bases de datos. BKS y PFM presentan los mayores valores de Accuracy debido a que realizan una mejor clasificación de las clases más representadas en la base de datos. Estas clases cuentan con un número mayor de muestras en la base de datos, lo que conlleva mayor número de posibles aciertos, y por tanto, mayor valor de Accuracy.

En el extremo opuesto se sitúa el método TRURG, que obtiene mejores resultados en la mayoría de bases de datos (8 de 12 casos) en lo que a las medidas de UAR, MAUC y Fscore se refiere, debido a que realiza una mejor identificación de las muestras de las clases menos representadas en la base de datos. Consigue así una mejor clasificación general, sin embargo, penaliza el valor de Accuracy al cometer mayor número de errores en las clases mayoritarias, y por tanto, mayor error global.

Tabla 4.3: Resultados obtenidos al aplicar los métodos de fusión de etiquetas a los clasificadores "sencillos" utilizando bases de datos desequilibradas (de nuevo en negrita el mejor resultado en cada caso y en cursiva los valores que no presentan diferencias estadísticamente significativas con éste, ambos casos en verde)

		ORA	MAJ	BBI	PFM	BKS	TRG			ORA	MAJ	BBI	PFM	BKS	TRG
Magic	Acc.	0.920	0.830	0.830	*0.837*	**0.838**	0.795	Pima	Acc.	0.881	**0.763**	*0.762*	0.756	0.755	0.752
	UAR	0.888	0.777	0.777	*0.805*	**0.806**	0.793		UAR	0.852	0.719	0.689	0.710	0.710	**0.733**
	Gmean	0.882	0.758	0.758	**0.799**	*0.798*	0.793		Gmean	0.846	0.703	0.649	0.692	0.695	**0.730**
	MAUC	0.888	0.777	0.777	*0.805*	**0.806**	0.793		MAUC	0.852	0.719	0.689	0.710	0.710	**0.733**
	Fscore	0.912	0.808	0.808	**0.817**	**0.817**	0.784		Fscore	0.865	**0.729**	0.719	0.721	0.719	**0.729**
Thyroid	Acc.	0.985	0.894	0.928	*0.946*	*0.944*	**0.947**	Dermatology	Acc.	0.977	0.921	*0.932*	0.930	*0.934*	**0.943**
	UAR	0.966	0.802	0.866	0.900	0.900	**0.924**		UAR	0.962	0.894	*0.915*	0.912	*0.920*	**0.926**
	Gmean	0.964	0.764	0.843	0.892	0.893	**0.919**		Gmean	0.944	0.851	*0.885*	0.890	*0.901*	**0.907**
	MAUC	0.974	0.863	0.902	0.925	0.924	**0.942**		MAUC	0.987	0.964	*0.968*	0.964	*0.968*	**0.972**
	Fscore	0.978	0.858	0.901	0.925	0.924	**0.931**		Fscore	0.970	0.913	*0.925*	0.923	*0.928*	**0.936**
Balance	Acc.	0.897	0.795	0.833	**0.858**	0.850	0.824	Glass	Acc.	0.793	0.554	0.583	*0.588*	0.579	**0.604**
	UAR	0.659	0.575	0.598	*0.617*	**0.619**	0.599		UAR	0.700	0.479	0.477	0.498	0.501	**0.548**
	Gmean	0.226	0.083	0.015	0.027	**0.144**	*0.126*		Gmean	0.318	0.055	0.023	0.034	0.054	**0.130**
	MAUC	0.806	0.674	0.724	**0.759**	*0.757*	0.722		MAUC	0.891	0.792	0.769	0.774	0.769	**0.812**
	Fscore	0.696	0.568	0.579	0.598	**0.609**	0.588		Fscore	0.720	0.451	0.461	0.483	0.495	**0.521**
Soybean	Acc.	0.933	0.851	0.875	0.867	*0.880*	**0.885**	Lymphography	Acc.	0.896	0.774	*0.788*	*0.791*	**0.792**	*0.789*
	UAR	0.923	0.837	*0.868*	0.856	**0.878**	**0.878**		UAR	0.706	**0.485**	0.416	*0.446*	*0.458*	*0.476*
	Gmean	0.740	0.534	0.583	0.636	**0.755**	*0.646*		Gmean	0.189	0.000	0.000	0.018	0.027	0.018
	MAUC	0.982	0.961	0.960	0.949	0.954	**0.965**		MAUC	0.809	**0.651**	0.588	0.611	*0.620*	*0.634*
	Fscore	0.932	0.858	*0.881*	0.871	*0.889*	**0.890**		Fscore	0.679	**0.475**	0.414	*0.439*	*0.448*	*0.468*
Ecoli	Acc.	0.837	0.741	0.731	0.744	0.744	**0.767**	Anneal	Acc.	0.985	0.955	0.957	**0.963**	**0.963**	0.957
	UAR	0.480	0.381	0.371	0.384	0.391	**0.417**		UAR	0.889	**0.823**	0.734	0.785	0.794	*0.812*
	Gmean	0.000	0.000	0.000	0.000	0.000	0.000		Gmean	0.730	**0.574**	0.000	0.256	0.230	0.368
	MAUC	0.704	0.672	0.673	0.671	*0.677*	**0.682**		MAUC	0.947	**0.902**	0.837	0.875	0.872	0.888
	Fscore	0.502	0.384	0.373	0.387	0.397	**0.426**		Fscore	0.914	**0.839**	0.749	0.803	0.793	0.810
Yeast	Acc.	0.727	0.549	0.553	0.560	0.555	**0.570**	Hypothyroid	Acc.	0.997	0.988	0.984	0.988	0.988	**0.990**
	UAR	0.550	0.403	0.368	0.390	0.378	**0.441**		UAR	0.731	0.691	0.658	0.686	0.687	**0.711**
	Gmean	0.000	0.000	0.000	0.000	0.000	0.000		Gmean	0.000	0.000	0.000	0.000	0.000	0.000
	MAUC	0.774	*0.722*	0.691	0.712	0.699	**0.729**		MAUC	0.821	0.763	0.711	0.750	0.757	**0.802**
	Fscore	0.593	0.413	0.382	0.407	0.399	**0.443**		Fscore	0.735	0.699	0.687	0.698	0.700	**0.706**

La mayoría de estos resultados obtenidos por el método TRURG presentan además diferencias estadísticamente significativas con los del resto de métodos analizados (6, 5 y 5 casos de 8 respectivamente para medidas de UAR, MAUC y Fscore) en base a las pruebas efectuadas mediante el test Wilcoxon.

En cuanto al resto de algoritmos analizados, se puede observar que el método BBI ha obtenido los peores resultados generales en este caso. El método MAJ por el contrario, ha obtenido mejores resultados que en el experimento con bases de datos equilibradas, con valores cercanos al método propuesto en algunos casos.

Por último, cabe resaltar que los experimentos realizados muestran cómo la medida Gmean puede no ser apropiada en casos en los que se utiliza una base de datos especialmente compleja (con muy pocas muestras en alguna de las clases). Si se clasifican de forma errónea todas las muestras de una de las clases de la base de datos se obtiene un valor de Gmean de 0, incluso si el resto de muestras pertenecientes al resto de clases son clasificadas de forma perfecta, por lo que no es posible extraer información acerca de la clasificación general realizada por los distintos métodos analizados en estos casos. A pesar de ello, en aquellos casos en los que la mayor parte de los métodos analizados han obtenido un valor de Gmean diferente de 0, el algoritmo TRURG es de nuevo el que obtiene mejores resultados en mayor número de las bases de datos utilizadas (4 casos de 8; 3 de los cuales presentan diferencias estadísticamente significativas con el resto de métodos).

Tras los resultados obtenidos, se han realizado nuevos experimentos para evaluar la robustez del método propuesto respecto a los clasificadores originales seleccionados. Se han seleccionado para proporcionar las etiquetas tres nuevos clasificadores más complejos que los utilizados en el experimento anterior, Bosques por rotación (Rotation Forest) [117], SVM [107] y MP [119], entrenados por medio de WEKA. Se han utilizado las mismas bases de datos y procedimiento del experimento previo. A continuación se muestran los nuevos resultados obtenidos.

En la Tabla 4.4 se muestra el rendimiento de los distintos métodos al utilizar bases de datos equilibradas. Se puede observar cómo los resultados obtenidos son similares a los recogidos en el experimento anterior (con clasificadores más sencillos). Los métodos BKS y PFM son los métodos que muestran un mejor rendimiento de acuerdo con los valores obtenidos de Accuracy, UAR, Fscore y Gmean, aunque no presentan diferencias estadísticamente significativas en gran parte de los

Tabla 4.4: Resultados obtenidos al aplicar los distintos métodos de fusión de etiquetas a los clasificadores complejos utilizando bases de datos equilibradas (en negrita el mejor resultado en cada caso y en cursiva los valores que no presentan diferencias estadísticamente significativas con éste, ambos casos en verde)

		ORA	MAJ	BBI	PFM	BKS	TRG			ORA	MAJ	BBI	PFM	BKS	TRG
Vowel	Acc.	0.862	0.722	*0.749*	*0.752*	**0.755**	*0.750*	Segment	Acc.	0.980	0.955	0.958	**0.962**	*0.960*	0.958
	UAR	0.867	0.730	*0.756*	*0.760*	**0.762**	*0.757*		UAR	0.980	0.956	0.958	**0.962**	*0.961*	0.958
	Gmean	0.860	0.711	*0.736*	*0.744*	**0.750**	*0.745*		Gmean	0.980	0.954	0.957	**0.961**	*0.960*	0.957
	MAUC	0.963	**0.932**	0.928	0.926	0.918	*0.930*		MAUC	0.993	**0.988**	**0.988**	**0.988**	**0.988**	**0.988**
	Fscore	0.871	0.734	*0.756*	*0.760*	**0.762**	0.756		Fscore	0.980	0.956	0.958	**0.962**	*0.961*	0.958
Waveform	Acc.	0.917	**0.848**	**0.848**	0.843	0.843	**0.848**	Pendigits	Acc.	0.994	0.984	0.985	**0.986**	**0.986**	0.985
	UAR	0.917	**0.848**	**0.848**	0.843	0.843	**0.848**		UAR	0.994	0.984	0.985	**0.986**	**0.986**	0.985
	Gmean	0.916	**0.848**	**0.848**	0.842	0.843	**0.848**		Gmean	0.994	0.984	0.985	**0.986**	**0.986**	0.985
	MAUC	0.942	**0.894**	**0.894**	0.890	0.890	**0.894**		MAUC	0.998	**0.995**	**0.995**	**0.995**	**0.995**	**0.995**
	Fscore	0.917	**0.848**	**0.848**	0.843	0.843	**0.848**		Fscore	0.994	0.984	0.985	*0.986*	**0.987**	0.985
Vehicle	Acc.	0.899	**0.776**	0.766	0.769	0.759	**0.776**	Letters	Acc.	0.934	0.876	0.874	**0.881**	0.878	0.875
	UAR	0.895	**0.768**	0.759	*0.762*	0.752	**0.768**		UAR	0.933	0.875	0.874	**0.881**	0.877	0.874
	Gmean	0.887	*0.737*	0.696	*0.728*	0.724	**0.743**		Gmean	0.932	0.873	0.872	**0.880**	0.876	0.873
	MAUC	0.956	**0.904**	0.899	0.900	0.893	*0.902*		MAUC	0.981	**0.966**	0.965	0.963	0.958	0.965
	Fscore	0.897	*0.767*	*0.758*	*0.764*	0.756	**0.768**		Fscore	0.935	0.878	0.876	**0.882**	0.878	0.876

casos. De nuevo los métodos TRURG y BBI muestran resultados muy similares, siendo MAJ el algoritmo con peor rendimiento.

En la Tabla 4.5 se muestra el rendimiento de los métodos al utilizar bases de datos desequilibradas. Los resultados obtenidos están altamente correlados con los del experimento anterior utilizando clasificadores más sencillos. Los métodos BKS y PFM obtienen un elevado valor de Accuracy al realizar una mejor clasificación de las clases más representadas en la base de datos. De nuevo el algoritmo TRURG consigue los mayores valores de UAR, MAUC, y Fscore en la mayoría de casos, 9 de 12, aunque no todos muestran diferencias estadísticamente significativas, en particular en los valores de Fscore (6, 4 y 2 casos de 9 son estadísticamente significativas para UAR, MAUC, y Fscore respectivamente). Al utilizar mejores clasificadores el potencial de mejora se ve reducido, ya que la mayor parte de las muestras se encuentran ya correctamente etiquetadas, por lo que resulta mas difícil encontrar diferencias en el rendimiento de los distintos métodos. Sin embargo, estas diferencias son notorias en los valores de UAR y Gmean, donde el algoritmo TRURG obtiene mejores resultados que el resto de métodos analizados en el estudio.

Tabla 4.5: Resultados obtenidos al aplicar los métodos de fusión de etiquetas a los clasificadores complejos utilizando bases de datos desequilibradas (de nuevo en negrita el mejor resultado en cada caso y en cursiva los valores que no presentan diferencias estadísticamente significativas con éste, ambos casos en verde)

		ORA	MAJ	BBI	PFM	BKS	TRG
Magic	Acc.	0.917	0.855	0.855	**0.860**	**0.860**	0.857
	UAR	0.891	0.817	0.817	0.829	0.828	**0.835**
	Gmean	0.887	0.808	0.808	0.823	0.822	**0.832**
	MAUC	0.891	0.817	0.817	0.829	0.828	**0.835**
	Fscore	0.908	0.836	0.836	**0.843**	*0.842*	0.841
Thyroid	Acc.	0.960	0.919	*0.936*	*0.936*	*0.937*	**0.938**
	UAR	0.905	0.837	*0.876*	*0.874*	*0.878*	**0.883**
	Gmean	0.897	0.816	*0.866*	0.863	*0.869*	**0.874**
	MAUC	0.933	0.885	*0.911*	*0.910*	*0.912*	**0.915**
	Fscore	0.941	0.892	*0.914*	*0.913*	*0.915*	**0.917**
Balance	Acc.	0.937	*0.889*	0.887	**0.892**	**0.892**	0.886
	UAR	0.795	0.692	0.650	0.732	*0.736*	**0.747**
	Gmean	0.735	0.535	0.188	0.620	0.649	**0.679**
	MAUC	0.873	0.824	0.802	0.830	0.833	**0.843**
	Fscore	0.868	0.727	0.654	*0.744*	*0.745*	**0.749**
Soybean	Acc.	0.931	0.883	0.882	**0.888**	*0.887*	*0.887*
	UAR	0.953	0.920	*0.920*	*0.924*	*0.924*	**0.927**
	Gmean	0.882	0.842	0.777	*0.821*	*0.823*	**0.855**
	MAUC	0.990	**0.982**	*0.980*	0.979	0.978	*0.981*
	Fscore	0.955	*0.924*	*0.922*	*0.926*	*0.926*	**0.928**
Ecoli	Acc.	0.881	0.799	0.782	0.793	0.792	**0.821**
	UAR	0.599	0.493	0.463	0.473	0.477	**0.532**
	Gmean	0.000	0.000	0.000	0.000	0.000	0.000
	MAUC	0.779	*0.735*	0.718	0.718	0.720	**0.742**
	Fscore	0.623	0.518	0.479	0.496	0.500	**0.555**
Yeast	Acc.	0.730	0.572	0.564	0.565	0.558	**0.583**
	UAR	0.546	0.420	0.380	0.393	0.379	**0.447**
	Gmean	0.011	0.000	0.000	0.000	0.003	0.000
	MAUC	0.763	*0.734*	0.701	0.710	0.698	**0.738**
	Fscore	0.611	0.444	0.405	0.420	0.406	**0.454**

		ORA	MAJ	BBI	PFM	BKS	TRG
Pima	Acc.	0.863	**0.766**	**0.766**	*0.762*	0.758	0.753
	UAR	0.825	0.713	0.699	0.710	0.703	**0.729**
	Gmean	0.817	0.694	0.667	0.689	0.681	**0.723**
	MAUC	0.825	0.713	0.699	0.710	0.703	**0.729**
	Fscore	0.843	**0.728**	*0.724*	*0.726*	0.719	*0.727*
Dermatology	Acc.	0.980	*0.958*	**0.961**	*0.959*	*0.960*	*0.959*
	UAR	0.981	*0.958*	**0.962**	*0.959*	*0.961*	*0.961*
	Gmean	0.980	*0.955*	**0.960**	*0.956*	*0.959*	*0.959*
	MAUC	0.994	**0.986**	**0.986**	*0.985*	*0.985*	**0.986**
	Fscore	0.982	*0.960*	**0.963**	*0.961*	**0.963**	*0.962*
Glass	Acc.	0.800	*0.631*	**0.637**	*0.628*	*0.626*	*0.636*
	UAR	0.702	0.532	0.481	0.482	0.493	**0.553**
	Gmean	0.478	*0.198*	0.041	0.050	0.084	**0.259**
	MAUC	0.866	*0.778*	0.747	0.742	0.748	**0.785**
	Fscore	0.759	*0.543*	0.484	0.496	0.504	**0.555**
Lymphography	Acc.	0.895	*0.801*	*0.803*	**0.806**	*0.805*	*0.801*
	UAR	0.731	**0.600**	0.488	*0.562*	0.500	0.535
	Gmean	0.285	0.089	0.000	0.072	0.009	0.000
	MAUC	0.821	**0.727**	0.626	*0.695*	0.633	0.660
	Fscore	0.718	**0.576**	0.480	*0.539*	0.491	*0.526*
Anneal	Acc.	0.986	0.974	0.971	**0.978**	0.976	0.972
	UAR	0.885	**0.858**	0.779	*0.855*	0.810	*0.845*
	Gmean	0.660	**0.628**	0.093	*0.553*	0.221	0.457
	MAUC	0.947	**0.925**	0.868	*0.922*	0.892	*0.918*
	Fscore	0.907	**0.883**	0.788	*0.877*	0.816	0.850
Hypothyroid	Acc.	0.991	0.959	0.957	**0.986**	**0.986**	0.983
	UAR	0.699	0.540	0.515	0.677	0.675	**0.686**
	Gmean	0.000	0.000	0.000	0.000	0.000	0.000
	MAUC	0.739	0.666	0.648	0.738	0.734	**0.751**
	Fscore	0.718	0.607	0.579	**0.690**	**0.690**	*0.684*

4.5.3 Reconocimiento de emociones a partir de la voz

A pesar de los buenos resultados obtenidos por el algoritmo propuesto en condiciones de desequilibrio en la base de datos, las diferencias existentes con el resto de métodos no son siempre estadísticamente significativas (probablemente debido al reducido numero de muestras y la poca complejidad de las bases de datos), por lo que no podemos considerar del todo concluyentes los experimentos realizados sobre las bases de datos del repositorio UCI. Con el fin de obtener una mejor comprensión acerca del comportamiento de los distintos algoritmos de fusión cuando se utilizan bases de datos desequilibradas, se han realizado nuevos experimentos en un problema ampliamente conocido en el ámbito en el procesamiento de voz: el reconocimiento de emociones a partir de la voz [29] [148] [25]. En esta tarea, las grabaciones de voz de distintos locutores en distintos idiomas deben ser categorizadas según diferentes emociones humanas. La expresión humana en el mundo real está dominada normalmente por el estado neutro, lo que a menudo deriva en bases de datos altamente desequilibradas y convierte el reconocimiento de emociones en un marco óptimo para la evaluación del algoritmo de fusión propuesto.

Concretamente se han utilizado diversos clasificadores desarrollados por el laboratorio Aholab con motivo de su participación en el Interspeech 2009 Emotion Challenge [125], así como la base de datos FAU Aibo Emotion Corpus [11] proporcionada por la organización del challenge. Del mismo modo, se ha conservado el reparto original de la base de datos entre entrenamiento y test propuesto para el challenge. La identificación de 5 clases no estandarizadas (enfado, enfático, neutro, positivo, otros) se propuso como tarea principal en este caso. La clase otros resulta de una agrupación heterogénea de las emociones no incluidas en el resto de categorías propuestas, por lo que no se ha tenido en cuenta a la hora de llevar a cabo los experimentos descritos a continuación. La distribución de la base de datos muestra el siguiente desequilibrio entre clases: enfado 5.43 %, enfático 7.60 %, neutro 83.24 % y positivo 3.73 %. Para mantener la uniformidad en los experimentos, la parte de entrenamiento de la base de datos definida originalmente en el challenge ha sido dividida aleatoriamente en dos bloques de igual tamaño. El primero de los bloques se ha destinado al entrenamiento de los clasificadores, mientras que el segundo se ha utilizado para construir las matrices de confusión y entrenar los

distintos métodos de fusión. Este proceso de división se ha realizado sobre 100 repeticiones y se han promediado los resultados obtenidos en cada repetición.

Se han llevado a cabo dos experimentos diferentes. En el primer experimento se han utilizado 3 clasificadores SVM entrenados con distintos tipos de parámetros estadísticos a largo plazo, a partir de los segmentos de voz: el primero basado en análisis prosódico de la señal de voz (PROS), el segundo por medio de parámetros relativos a la calidad de la voz (VQ) y el último a través de parámetros relativos a la distribución de la energía del espectro de la señal (SPEC). Los resultados obtenidos por cada uno de los clasificadores individuales en la parte de test de la base de datos están recogidos en la Tabla 4.6.

Posteriormente, se ha realizado un segundo experimento a partir de las etiquetas proporcionadas por tres clasificadores entrenados con el mismo tipo de parámetros (todos ellos basados en el espectro de la señal). De esta forma pretendemos evaluar el comportamiento de los distintos métodos al utilizar clasificadores entrenados con parámetros altamente correlados, ya que se puede considerar que dos clasificadores son independientes si utilizan distinta base de datos de entrenamiento o son entrenados con distinto tipo de parámetros [153]. El primer clasificador elegido ha sido de nuevo el SVM entrenado con parámetros relativos a la distribución de la energía del espectro de la señal (SPEC), el segundo ha sido un clasificador GMM entrenado con una versión a corto plazo de los mismos parámetros espectrales (LFPC) y el tercero un nuevo clasificador GMM entrenado con los mismos parámetros a corto plazo extraídos únicamente de las tramas sonoras (LFPC_V). Los resultados obtenidos por cada uno de los nuevos clasificadores en la parte de test de la base de datos se muestran en la Tabla 4.7.

A continuación se recogen los resultados obtenidos al aplicar los distintos métodos de fusión de etiquetas en los dos experimentos propuestos. Al igual que en el caso de los experimentos realizados sobre las bases de datos del repositorio UCI se muestra en negrita el mejor resultado en cada experimento y se presentan en cursiva los valores que no presentan diferencias estadísticamente significativas con éste. Se mantiene igualmente el color verde para mejorar la visualización de dichos valores. De nuevo se ha utilizado el test Wilcoxon con un nivel de significación del 95 % para realizar la comparación estadística entre los métodos distintos de fusión.

Tabla 4.6: Resultados de los clasificadores individuales entrenados con distinto tipo de parámetros sobre la parte de test de la base de datos AIBO

	PROS	VQ	SPEC
Accuracy	0.538	0.444	0.743
UAR	0.531	0.372	0.550
Gmean	0.511	0.336	0.520
MAUC	0.687	0.588	0.702
Fscore	0.414	0.322	0.463

Tabla 4.7: Resultados de los clasificadores individuales entrenados con parámetros espectrales sobre la parte de test de la base de datos AIBO

	LFPC	LFPC_V	SPEC
Accuracy	0.624	0.705	0.743
UAR	0.595	0.578	0.550
Gmean	0.576	0.553	0.520
MAUC	0.737	0.722	0.702
Fscore	0.457	0.462	0.463

Del mismo modo, se han incluido los resultados del sistema Oracle (ORA) como referencia en cada una de las las tablas.

En la tabla 4.8 se muestran los resultados obtenidos en el primer experimento, con clasificadores independientes (entrenados con distinto tipo de parámetros). Se puede observar cómo únicamente los métodos MAJ y TRURG consiguen mejorar el funcionamiento de los clasificadores originales (el valor de UAR fue la métrica definida por la organización del challenge). El algoritmo TRURG es el método que obtiene un mejor rendimiento en términos de UAR, Gmean, MAUC y Fscore, aunque el test de Wilcoxon no muestre diferencias estadísticamente significativas entre los métodos TRURG y MAJ en el caso de los valores de MAUC obtenidos.

El mayor valor de Accuracy se ha obtenido al aplicar el algoritmo BBI, que clasifica como Neutro todas las muestras contenidas en la parte de test de la base de datos. Del mismo modo, el método PFM obtiene un valor elevado de Accuracy

Tabla 4.8: Resultados obtenidos al aplicar los distintos métodos de fusión a los clasificadores entrenados con distinto tipo de parámetros sobre la base de datos AIBO (en negrita el mejor resultado en cada caso y en cursiva los valores que no presentan diferencias estadísticamente significativas con éste, ambos casos en verde)

	ORA	MAJ	BBI	PFM	BKS	TRG
Accuracy	0.850	0.683	**0.877**	0.871	0.833	0.689
UAR	0.744	0.551	0.250	0.304	0.442	**0.556**
Gmean	0.721	0.522	0.000	0.012	0.353	**0.529**
MAUC	0.811	*0.703*	0.500	0.543	0.627	**0.706**
Fscore	0.636	0.449	0.234	0.336	0.439	**0.452**

al asignar la clase Neutro a la mayor parte de las muestras del test. Ambos métodos obtienen resultados muy pobres en las clases menos representadas, por lo que sus valores de UAR, Fscore, Gmean y MAUC son inferiores a los obtenidos por el resto de métodos utilizados. Este hecho no hace sino confirmar que el valor de Accuracy no es un parámetro adecuado para analizar el rendimiento de un clasificador en casos de alto grado de desequilibrio en el número de muestras pertenecientes a cada clase, como ocurre en la base de datos AIBO utilizada en este trabajo.

El método BKS etiqueta correctamente las muestras de las clases menos representadas en los casos más consistentes, por lo que consigue altos valores de Precisión para dichas clases. Sin embargo, obtiene pobres valores de UAR al perder muchas de las muestras de las clases minoritarias asignando de forma sistemática el estado Neutro a las muestras analizadas, lo que conlleva bajos valores de UAR, Fscore, Gmean y MAUC. Por el contrario, el método TRURG penaliza el valor de Precisión para las clases menos representadas al seleccionarlas más frecuentemente, por lo que pierde un número considerablemente menor de muestras de dichas clases y obtiene valores más elevados de UAR, Fscore, Gmean y MAUC que el resto de métodos (con menor valor de Fscore que el mejor clasificador original).

La Tabla 4.9 muestra resultados muy similares para el segundo experimento realizado, donde los clasificadores utilizados no son estadísticamente independientes (todos ellos han sido entrenados con parámetros espectrales extraídos de los segmentos de audio). De nuevo únicamente los métodos MAJ y TRURG son ca-

Tabla 4.9: Resultados obtenidos al aplicar los distintos métodos de fusión a los clasificadores entrenados con parámetros espectrales sobre la base de datos AIBO (nuevamente en negrita el mejor resultado en cada caso y en cursiva los valores que no presentan diferencias estadísticamente significativas con éste, ambos casos en verde)

	ORA	MAJ	BBI	PFM	BKS	TRG
Accuracy	0.859	0.720	**0.877**	0.856	0.841	0.716
UAR	0.724	*0.601*	0.250	0.422	0.511	**0.603**
Gmean	0.699	*0.578*	0.000	0.108	0.441	**0.580**
MAUC	0.796	0.738	0.500	0.631	0.673	**0.743**
Fscore	0.629	**0.481**	0.234	0.419	0.493	**0.481**

paces de mejorar los resultados obtenidos por los clasificadores originales, siendo el algoritmo TRURG el método que consigue mejores resultados en términos de UAR, Gmean, MAUC y Fscore, aunque el test de Wilcoxon no muestra diferencias significativas en lo que al valor de MAUC se refiere.

Al igual que ocurría en el experimento anterior, los métodos BBI y PFM han obtenido los valores más elevados de Accuracy y los menores de UAR, Fscore, Gmean y MAUC al seleccionar repetidamente el estado Neutro como salida para las nuevas muestras (todas las muestras del test en el caso del modo BBI).

El método TRURG consigue el valor de UAR más elevado al asignar de forma más asidua las clases menos representadas en detrimento de la clase Neutro. En este caso, no sólo los valores de UAR, Gmean y MAUC, sino también el valor de Fscore obtenidos resultan considerablemente superiores a los conseguidos por el clasificador original con mejor rendimiento.

Este experimento demuestra además cómo a pesar de necesitar (de forma teórica) independencia estadística entre los clasificadores originales, el método propuesto puede ser aplicado en la práctica en casos en los que dicha independencia no está asegurada, como ha ocurrido ya con otros algoritmos de fusión [80], [3].

Las diferencias existentes en el procedimiento de asignación de etiquetas de los distintos métodos resultan más evidentes en la Figura 4.2, donde se muestran las matrices de confusión obtenidas por los métodos TRURG y BKS en el primer experimento realizado (clasificadores entrenados con distinto tipos de parámetros). Al no disponer de un único resultado, se ha presentado el valor medio obtenido

28.13	8.63	6.22	50.02	0.302
4.24	163.97	64.47	59.32	0.562
3.99	38.15	243.45	85.41	0.656
151.86	590.75	843.72	3790.67	0.705
0.149	0.205	0.210	0.951	0.689

(a) Matriz de confusión TRURG

16.67	4.38	2.67	69.28	0.179
2.44	109.73	25.96	153.87	0.376
2.37	16.88	114.52	237.23	0.309
54.46	226.64	230.12	4865.78	0.905
0.220	0.307	0.307	0.914	0.833

(b) Matriz de confusión BKS

Figura 4.2: Matriz de confusión obtenidas por los algoritmos (a) TRURG y (b) BKS obtenidas en el primer experimento realizado en reconocimiento de emociones

para cada elemento de la matriz en cada caso. El método BKS ha sido elegido para llevar a cabo la comparación, ya que realiza una mejor clasificación de las clases minoritarias que los métodos BBI y PFM. Recordemos que las filas representan las clases reales a las que pertenecen las muestras del test, mientras que las columnas recogen las etiquetas asignadas por los algoritmos de fusión a dichas muestras. Los valores en color gris claro muestran el número de aciertos para cada una de las clases. Los valores en blanco muestran los errores cometidos por los algoritmos debidos a una mala clasificación. La última fila resume los resultados obtenidos en base a las predicciones realizadas por los algoritmos, es decir, los valores de Precisión obtenidos para cada clase individualmente. Del mismo modo, la última columna recoge los resultados obtenidos en base a la distribución de clases de la base de datos, es decir, los valores de Recall por clase. Por último, los valores en negro muestran el resultado general o Accuracy obtenido por los métodos.

El modo BKS obtiene mayores valores de Precisión para las clases menos representadas, ya que son seleccionadas en muy pocos casos, los más consistentes. Sin embargo, los valores obtenidos están lejos de ser óptimos (0.220, 0.307 y 0.307). La mayor parte de las muestras de las clases minoritarias son clasificadas como Neutro, lo que reduce el valor de Precisión para esta clase. Los errores por pérdidas son frecuentes salvo para la clase Neutro (que contiene la mayor parte de muestras de la base de datos), lo que resulta en un elevado valor de Accuracy en este caso.

El método TRURG, sin embargo, obtiene valores más elevados de Recall para las clases minoritarias, ya que son seleccionadas más frecuentemente (0.302, 0.562 y 0.656 respectivamente). Por el contrario, gran parte de las muestras de la clase

Neutro se clasifican erróneamente, lo que reduce el valor de Recall para esta clase. El valor de Precisión de la clase la clase mayoritaria es considerablemente más elevado en este caso, ya que sólo los casos más consistentes reciben esta etiqueta a la salida. En resumen, el método propuesto penaliza los resultados de la clase más representada en favor de las clases minoritarias, que son las que habitualmente contienen la información "útil" en este tipo de bases de datos.

4.5.4 Segmentación de audio

Una vez comprobada la eficacia del método de fusión propuesto en condiciones de desequilibrio en la base de datos utilizada, se han realizado diversos experimentos en un área concreta del procesado de señal de voz: la segmentación de audio.

Como se ha visto en el Capítulo 3, la segmentación de audio es una técnica de procesado ampliamente utilizada como paso previo en diversos sistemas de reconocimiento más complejos, por lo que el desarrollo de mejores clasificadores de audio resulta una idea de gran interés. Una alternativa sencilla es la de aprovechar el conocimiento de distintos clasificadores con la finalidad de obtener un mejor resultado global que al aplicarlos individualmente.

El limitado número de clases definidas tradicionalmente como objetivo en la segmentación de audio favorece además la utilización de alguna de las múltiples técnicas de fusión de clasificadores existentes, en especial aquellas que trabajan a nivel de etiqueta y que permiten una fácil integración de cualquier tipo de clasificador desarrollado por uno o incluso varios laboratorios de forma independiente.

Como es habitual en cualquiera de las distintas tecnologías del habla, existe un alto grado de desequilibrio entre las clases presentes en las bases de datos disponibles en el ámbito de la segmentación de audio, por lo que la técnica de fusión propuesta se presenta como una alternativa viable a los métodos de fusión de etiquetas tradicionales, que como se ha demostrado en la sección anterior, reducen considerablemente su rendimiento en este tipo de entornos.

En este sentido, se han realizado diversos experimentos para evaluar el comportamiento del algoritmo propuesto en este campo, utilizando para ello los datos y clasificadores desarrollados con motivo de la realización de las campañas de evaluación Albayzin 2012 y Albayzin 2014, organizadas por la RTTH.

4.5.4.1 Evaluación de segmentación de audio Albayzin 2012

Recordemos que el objetivo de la campaña de evaluación de sistemas de segmentación de audio Albayzin 2012 consistía en la segmentación de audio *broadcast*, asignando a cada uno de los distintos segmentos analizados etiquetas para indicar la presencia de voz, música y ruido, pudiendo existir solapamiento entre las tres clases en cualquier instante.

La base de datos proporcionada por la organización, anteriormente descrita, está formada por unas 20 horas de audio del canal de radio Aragón Radio con la siguiente distribución de clases: 22 % de voz limpia, 9 % de música, 31 % de voz con música de fondo, 26 % de voz con ruido de fondo y 12 % de otros. Para realizar el entrenamiento de los sistemas se permite además la utilización de la base de datos proporcionada en la campaña Albayzin 2010, formada por unas 87 horas de audio del canal de televisión 3/24 con la siguiente distribución: 37 % de voz limpia, 5 % de música, 15 % de voz con música de fondo, 40 % de voz con ruido de fondo y 3 % de otros. Se puede observar cómo ambas bases de datos muestran el desequilibrio descrito en la introducción de esta sección.

Como se ha comentado previamente en el Capítulo 3, la métrica escogida para evaluar el funcionamiento de los sistemas fue el SER, calculado como la suma de tres tipos de errores: el porcentaje de tiempo que es asignado a una clase incorrecta, el porcentaje de tiempo en el que una clase está presente pero no ha sido etiquetada y el porcentaje de tiempo en que se ha etiquetado una clase cuando realmente no estaba presente. Todos ellos obtenidos mediante las herramientas de evaluación proporcionadas por el NIST [98].

La tabla 4.10 recoge los resultados obtenidos por los 6 sistemas que tomaron parte en la campaña de evaluación Albayzin 2012. Se puede observar cómo el sistema diseñado por el grupo Aholab, descrito previamente en la sección 3.3.1.1, obtuvo el menor valor de SER, y se proclamó por tanto mejor sistema de la evaluación. En cuanto al resto de sistemas presentados nos referiremos a ellos con los nombres ficticios S2-S6 en atención al puesto que ocuparon en la evaluación.

El acceso a las marcas proporcionadas por los sistemas S3 y S6 ha determinado los primeros experimentos de fusión de sistemas de segmentación de audio. En primer lugar, con el fin de comprobar los resultados cuando dos clasificadores presentan bajas tasas de error, se han utilizado dos de los sistemas que mejores

Tabla 4.10: Resultados en términos de SER de los sistemas de segmentación de audio presentados en la campaña de Albayzin 2012

Sistema	SER (Test)
AHOLAB-EHU	**25.78 %**
S2	26.53 %
S3	28.12 %
S4	33.30 %
S5	39.55 %
S6	40.01 %

resultados obtuvieron en la evaluación, el sistema Aholab y el sistema S3. En segundo lugar, para comprobar la robustez del método de fusión propuesto se han seleccionado los sistemas que mejor y peor resultados obtuvieron en la evaluación Albayzin 2012, el sistema Aholab y el sistema S6.

Para llevar a cabo la fusión de los sistemas de segmentación de audio se ha realizado en primer lugar un mapeado de las clases para evitar el solapamiento de las mismas [140]. Para poder aplicar el método propuesto, la salida de cada sistema debe ser única en cada segmento, por lo que las clases originales (silencio, voz, música y ruido), han sido sustituidas por las diferentes combinaciones posibles entre ellas: silencio, voz limpia, música, ruido, voz con música, voz con ruido, música con ruido, voz con música y ruido.

A continuación, se han calculado las matrices de confusión de los sistemas implicados usando la parte de entrenamiento de la base de datos de Aragón Radio. Posteriormente se ha realizado la combinación de las salidas de los sistemas originales mediante la técnica de fusión propuesta y se han obtenido las marcas finales para cada segmento deshaciendo el mapeado realizado inicialmente.

Presentado el procedimiento, se muestran en las siguientes tablas los resultados obtenidos en los dos experimentos propuestos previamente. En primer lugar, la tabla 4.11 recoge el resultado, en términos de SER, de la fusión entre el sistema propuesto por el grupo Aholab y el sistema S3 (tercer mejor sistema de la campaña de evaluación de Albayzin 2012). Como se ha comentado anteriormente, el objetivo

Tabla 4.11: Resultado de la fusión de los sistemas Aholab y S3

Sistema	Entrenamiento	Test
AHOLAB	19.97 %	25.78 %
S3	21.24 %	28.12 %
Fusión	**16.13 %**	**18.86 %**

de este experimento es comprobar el rendimiento de la técnica de fusión propuesta cuando los dos clasificadores originales presentan bajas tasas de error.

Se puede observar cómo en este caso, con buenos resultados de partida en ambos sistemas, el algoritmo desarrollado obtiene una reducción relativa del SER del 19.5 % en la parte de entrenamiento y del 26.8 % en la parte de test respecto al mejor de los sistemas, lo que demuestra claramente la validez del método de fusión propuesto. No resultan obvias, sin embargo, las razones que impulsan la mejora obtenida al aplicar el algoritmo de fusión aplicado.

Con el fin de analizar el origen de la mejora de los resultados, se ha evaluado individualmente el error cometido para cada evento original definido originalmente en la evaluación (voz, música y ruido). Se incluyen además los errores de omisión y de inserción de cada clase, referidos al tiempo total asignado a dicha clase en la referencia, tal y como son calculados por las herramientas de evaluación de NIST.

La tabla 4.12 muestra el resultado obtenido para la clase de voz en la partes de entrenamiento y de test de la base de datos. Se puede observar cómo se obtiene una mejora considerable, a pesar de los buenos resultados para esta clase en cada uno de los sistemas originales, con un SER en torno al 4 % en el sistema Aholab. La distinta clasificación llevada a cabo por los dos sistemas de forma individual permite obtener un mejor resultado final.

El resultado obtenido para la clase de música se muestra en la tabla 4.13. En este caso se consigue una reducción del SER del 40 % en la parte de test de la base de datos. El algoritmo desarrollado permite utilizar las diferencias existentes entre la clasificación realizada por los sistemas para esta clase (cuatro clases tras el mapeado realizado: música, voz con música, música con ruido y música con voz y ruido) para mejorar el resultado final.

Tabla 4.12: Detalle del error cometido por los sistemas Aholab y S3 y la fusión de ambos para la clase 'voz'de manera individual

Sistema		Omisión	Inserción	SER
AHOLAB	Entrenamiento	3.9 %	0.8 %	4.63 %
S3		0.1 %	5.5 %	5.65 %
Fusión		**1.5 %**	**0.8 %**	**2.32 %**
AHOLAB	Test	3.3 %	0.9 %	4.19 %
S3		0.2 %	8.5 %	8.69 %
Fusión		**1.7 %**	**1.4 %**	**3.11 %**

Tabla 4.13: Detalle del error cometido por los sistemas Aholab y S3 y la fusión de ambos para la clase 'música'de manera individual

Sistema		Omisión	Inserción	SER
AHOLAB	Entrenamiento	26.8 %	4.1 %	30.86 %
S3		25.9 %	4.1 %	29.93 %
Fusión		**10.3 %**	**6.3 %**	**16.59 %**
AHOLAB	Test	36.9 %	6.7 %	43.59 %
S3		37.5 %	5.4 %	42.91 %
Fusión		**19.2 %**	**6.7 %**	**25.92 %**

Por último, la tabla 4.14 muestra el resultado obtenido para la clase de ruido. Se puede observar cómo, al igual que en el caso de las clases de voz y música, se ha conseguido una importante reducción del SER, un 22 % en la parte de test. A pesar del elevado error obtenido por los dos sistemas originales en esta clase, ambos aportan información de utilidad en este caso, y el algoritmo de fusión propuesto es capaz de mejorar el resultado final.

Una vez realizado el estudio respecto a cada etiqueta, se puede observar cómo se ha obtenido una mejora considerable de los resultados en todas las clases, con una reducción importante del error de segmentación principalmente en la clase música, que fue la que más dificultades de clasificación planteó en la campaña

Tabla 4.14: Detalle del error cometido por los sistemas Aholab y S3 y la fusión de ambos para la clase 'ruido'de manera individual

Sistema		Omisión	Inserción	SER
AHOLAB	Entrenamiento	33.3 %	9.5 %	42.85 %
S3		14.3 %	20.8 %	35.09 %
Fusión		**30.3 %**	**4.1 %**	**34.42 %**
AHOLAB	Test	34.8 %	28.2 %	63.03 %
S3		19.0 %	65.9 %	84.87 %
Fusión		**29.5 %**	**19.6 %**	**49.07 %**

de evaluación Albayzin 2012. Esto demuestra claramente el buen rendimiento del método de fusión de etiquetas propuesto en el ámbito de la segmentación de audio.

Se muestran a continuación los resultados obtenidos al aplicar el método de fusión propuesto a los sistemas que mejor y peor resultados consiguieron en la evaluación Albayzin 2012, el sistema Aholab y el sistema S6. Se pretende en este caso comprobar la robustez del método de fusión propuesto. La tabla 4.15 muestra el resultado, en términos de SER, de la fusión de los dos sistemas seleccionados tanto en la parte de entrenamiento de la base de datos como en la de test.

Podemos observar cómo de nuevo se logra una mejora de los resultados obtenidos en ambos casos, con una reducción relativa del SER del 6.3 % en la parte de entrenamiento y del 6.28 % en la parte de test respecto al mejor de los sistemas. Se trata de una mejora significativa teniendo en cuenta que uno de los sistemas utilizados parte con un SER del 40 %, el mayor de la campaña de evaluación. El método

Tabla 4.15: Resultado de la fusión de los sistemas Aholab y S6

Sistema	Entrenamiento	Test
AHOLAB	19.97 %	25.78 %
S6	35.93 %	40.01 %
Fusión	**18.71 %**	**24.16 %**

Tabla 4.16: Detalle del error cometido por los sistemas Aholab y S6 y la fusión de ambos para la clase 'voz'de manera individual

Sistema		Omisión	Inserción	SER
AHOLAB	Entrenamiento	3.9 %	0.8 %	4.63 %
S6		0.8 %	1.6 %	2.34 %
Fusión		**4.3 %**	**0.4 %**	**4.12 %**
AHOLAB	Test	3.3 %	0.9 %	4.19 %
S6		0.6 %	3.0 %	3.56 %
Fusión		**3.6 %**	**0.5 %**	**4.12 %**

de fusión propuesto es capaz de obtener información suficiente de este sistema y de utilizarla para mejorar los resultados del sistema propuesto por Aholab, lo que muestra en gran medida la robustez del algoritmo desarrollado.

Al igual que en el experimento anterior, se ha estudiado el comportamiento de la fusión en cada una de las clases originales para analizar el origen de la mejora en los resultados. La tabla 4.16 muestra el resultado obtenido para la clase de voz en las partes de entrenamiento y de test de la base de datos. Se puede observar cómo el resultado de la fusión en este caso es prácticamente nulo, debido principalmente a los buenos resultados obtenidos para esta clase en cada uno de los sistemas originales, en particular el sistema S6 que presenta un SER en torno a sólo el 3 %.

El resultado obtenido para la clase de música se muestra en la tabla 4.17. Al igual que en el caso anterior, se consigue una reducción considerable del SER del 10 % en la parte de test de la base de datos. En este caso el error original de los sistemas es mayor (65.76 % de error para el sistema S6) y la mejora obtenida tras aplicar el algoritmo de fusión es menos elevada.

Por último, la tabla 4.18 muestra el resultado obtenido para la clase de ruido. Al igual que en el caso de la voz, el resultado de la fusión en este caso es casi inapreciable, debido principalmente a los resultados del sistema S6 con un 162.18 % de SER. En este caso resulta imposible extraer información de utilidad con la que mejorar los resultados. Sin embargo, cabe resaltar que el resultado final no se ve comprometido a pesar de estas tasas de error superiores al 100 %, manteniéndose

Tabla 4.17: Detalle del error cometido por los sistemas Aholab y S6 y la fusión de ambos para la clase 'música'de manera individual

Sistema		Omisión	Inserción	SER
AHOLAB	Entrenamiento	26.8 %	4.1 %	30.86 %
S6		56.6 %	1.8 %	58.41 %
Fusión		**25.4 %**	**4.5 %**	**29.94 %**
AHOLAB	Test	36.9 %	6.7 %	43.59 %
S6		64.3 %	1.4 %	65.76 %
Fusión		**33.9 %**	**5.4 %**	**39.34 %**

Tabla 4.18: Detalle del error cometido por los sistemas Aholab y S6 y la fusión de ambos para la clase 'ruido'de manera individual

Sistema		Omisión	Inserción	SER
AHOLAB	Entrenamiento	33.3 %	9.5 %	42.85 %
S6		52.7 %	61.3 %	113.95 %
Fusión		**36.0 %**	**7.5 %**	**43.57 %**
AHOLAB	Test	34.8 %	28.2 %	63.03 %
S6		42.5 %	119.6 %	162.18 %
Fusión		**38.8 %**	**24.8 %**	**63.61 %**

en el orden logrado por el mejor de los sistemas, lo que demuestra la robustez del método desarrollado. Al igual que en el experimento anterior, se puede observar cómo la mejora obtenida al aplicar el método de fusión propuesto se debe principalmente al resultado obtenido en la clase música, motivo de preocupación especial en la campaña Albayzin 2012 a la hora de diseñar los sistemas.

Para finalizar los experimentos relativos a la campaña de evaluación Albayzin 2012, se ha llevado a cabo un último experimento a partir de las etiquetas proporcionadas por los tres sistemas utilizados anteriormente. En este caso se pretende conseguir una mejora mayor en los resultados al utilizar de forma simultánea las salidas de los sistemas Aholab, S3 y S6, de forma que se pueda evaluar el máximo

Tabla 4.19: Resultado de la fusión de los sistemas Aholab, S3 y S6

Sistema	Entrenamiento	Test
AHOLAB	19.97 %	25.78 %
S3	21.24 %	28.12 %
S6	35.93 %	40.01 %
Fusión	**15.61 %**	**18.21 %**

potencial del algoritmo de fusión desarrollado en este ámbito. La tabla 4.19 muestra el resultado, en términos de SER, de la fusión de los tres sistemas seleccionados tanto en la parte de entrenamiento de la base de datos como en la de test.

Podemos observar cómo la mejora obtenida es mayor en ambos casos, con una reducción relativa del SER del 21.8 % en la parte de entrenamiento y del 29.4 % en la parte de test respecto al mejor de los sistemas. Se trata de una mejora significativa teniendo en cuenta el error obtenido por el sistema Aholab, mejor sistema en la campaña de evaluación con un 25.78 % de SER y el 18.21 % final de SER arrojado por el algoritmo de fusión propuesto.

Estos resultados demuestran claramente el buen rendimiento del método propuesto en el ámbito de la segmentación de audio, donde se muestra como una alternativa capaz de obtener la información útil aportada por cada sistema y de utilizarla para obtener una mejora realmente significativa de los resultados.

4.5.4.2 **Albayzin 2014**

El objetivo de la campaña de evaluación de sistemas de segmentación de audio Albayzin 2014 perseguía la segmentación de audio *broadcast*, asignando de nuevo a los segmentos de audio etiquetas para indicar la presencia de voz, música y ruido, pudiendo existir solapamiento entre las tres clases en cualquier instante.

La base de datos proporcionada por la organización, descrita en el Capítulo 2, incluye sonidos ambientales de diversas fuentes [46] [61] en las grabaciones ya utilizadas en las ediciones anteriores de 2010 y 2012. Fruto de esta combinación, se obtuvieron 35 nuevas sesiones (20 para entrenamiento y 15 para test) con el siguiente reparto de clases: 30 % de voz limpia, 4 % de música, 20 % de voz con música de fondo, 30 % de voz con ruido de fondo y 16 % de otros, lo que de nuevo refleja el desequilibrio habitual en las bases de datos utilizadas en este ámbito.

La métrica escogida para evaluar el funcionamiento de los sistemas fue nuevamente el SER, calculado al igual que en las ediciones anteriores mediante las herramientas de evaluación proporcionadas por el NIST [98].

Como se ha descrito previamente la sección 3.3.2, el sistema de segmentación de audio implementado con motivo de la campaña Albayzin 2014 incluye en su diseño la fusión de etiquetas de dos subsistemas (i-vectors y HMM) mediante el algoritmo TRURG, por lo que los experimentos realizados en este caso se han llevado a cabo como parte del desarrollo del propio sistema. Recordemos que el reparto de las sesiones de entrenamiento de la base de datos que se ha definido era el siguiente: 1-15 entrenamiento, 16-20 desarrollo.

El mismo procedimiento descrito en la Sección 4.5.4.1 se ha llevado a cabo a la hora realizar la fusión de las etiquetas proporcionadas por los dos subsistemas (mapeado de las clases, cálculo de las matrices de confusión de los subsistemas, combinación de etiquetas y mapeado inverso).

La tabla 4.20 recoge el resultado, en términos de SER, de la fusión entre los dos subsistemas en las sesiones de entrenamiento y desarrollo. Se puede observar cómo a pesar del bajo rendimiento del sistema HMM, se ha obtenido una reducción relativa del SER del 8 % en la parte de desarrollo respecto al sistema de i-vectors, lo que de nuevo demuestra la validez del método de fusión propuesto.

Al igual que en los experimentos anteriores, se ha evaluado el error cometido para cada evento original individualmente con el fin de analizar el origen de la

Tabla 4.20: SER obtenido en cada paso del sistema de segmentación de audio propuesto (sistema basado en HMM, sistema basado en i-vectors y fusión de los mismos) en las sesiones de entrenamiento y desarrollo de la base de datos Albayzin 2014

Ses.	HMM	i-vec	Fusión	Ses.	HMM	i-vec	Fusión
01	17.27 %	9.55 %	8.7 %	09	19.3 %	10.35 %	9.33 %
02	20.82 %	10.84 %	9.81 %	10	17.24 %	12.55 %	10.36 %
03	16.07 %	10.29 %	8.9 %	11	20.77 %	10.36 %	9.09 %
04	21.43 %	10.33 %	9.58 %	12	17.02 %	8.11 %	6.69 %
05	17.88 %	9.39 %	8.52 %	13	17.48 %	10.33 %	8.81 %
06	22.94 %	14.6 %	13.14 %	14	19.38 %	12.94 %	11.71 %
07	28.18 %	10.27 %	9.64 %	15	19.75 %	8.87 %	7.81 %
08	13.97 %	10.92 %	9.7 %				
1-15	19.38 %	10.67 %	**9.49 %**				
16	18.79 %	15.72 %	14.55 %	19	19.68 %	15.66 %	14.85 %
17	14.17 %	12.87 %	11.23 %	20	33.09 %	21.73 %	20.55 %
18	25.74 %	16.35 %	14.98 %				
16-20	21.99 %	16.33 %	**15.1 %**				

mejora de los resultados. Nuevamente se incluyen errores de omisión y de inserción de cada clase, referidos al tiempo total asignado a dicha clase en la referencia.

La tabla 4.21 muestra el resultado obtenido para la clase de voz en la parte de desarrollo la base de datos. Se puede observar cómo el efecto de la fusión es prácticamente despreciable en este caso, debido en parte a los buenos resultados para esta clase en cada uno de los sistemas originales, con un SER en torno al 5 % en el sistema HMM, el mejor para esta clase.

La tabla 4.22 muestra el resultado obtenido para la clase de música. En este caso se consigue una considerable reducción del SER del 18 % respecto al sistema de i-vectors. Al igual que en los experimentos anteriores, el algoritmo de fusión permite utilizar las diferencias existentes entre la clasificación realizada por los sistemas para esta clase para mejorar el resultado final.

Tabla 4.21: Detalle del error cometido por los dos subsistemas y la fusión de ambos para la clase 'voz'de manera individual

Sistema		Inserción	Omisión	SER
HMM	Desarrollo	2.0 %	3.3 %	5.3 %
i-vec		4.4 %	3.2 %	7.6 %
Fusión		**3.0 %**	**3.1 %**	**6.1 %**

Tabla 4.22: Detalle del error cometido por los dos subsistemas y la fusión de ambos para la clase 'música'de manera individual

Sistema		Inserción	Omisión	SER
HMM	Desarrollo	23.2 %	9.4 %	32.6 %
i-vec		11.9 %	6.5 %	18.4 %
Fusión		**8.6 %**	**6.4 %**	**15.0 %**

Por último, la tabla 4.23 muestra el resultado obtenido para la clase de ruido. Al igual que en el caso de la voz, el resultado de la fusión en este caso es reducido, debido principalmente a los malos resultados del sistema HMM con un 70 % de SER. En este caso resulta imposible extraer información de utilidad con la que mejorar los resultados. Al igual que en los experimentos anteriores, se puede observar cómo la mejora obtenida al aplicar el método de fusión propuesto se debe principalmente al resultado obtenido en la clase música.

Para finalizar los experimentos relativos a la campaña de evaluación Albayzin 2014 en particular y a la segmentación de audio en general, se muestra a continuación el resultado obtenido al aplicar el algoritmo de fusión propuesto a los dos subsistemas en la parte de test de la base de datos.

La tabla 4.24 muestra el resultado, en términos de SER. Podemos observar cómo la mejora conseguida es similar a la obtenida en la parte de desarrollo, con una reducción relativa del SER del 8 % respecto al sistema de i-vectors. Se trata de una mejora aceptable teniendo en cuenta el aumento del error en ambos subsistemas en la parte de test de la base de datos.

Tabla 4.23: Detalle del error cometido por los dos subsistemas y la fusión de ambos para la clase 'ruido'de manera individual

Sistema		Inserción	Omisión	SER
HMM	Desarrollo	28.9 %	41.2 %	70.11 %
i-vec		18.1 %	28.4 %	46.5 %
Fusión		**17.9 %**	**28.6 %**	**43.57 %**

Tabla 4.24: SER obtenido en cada paso del sistema de segmentación de audio propuesto (sistema basado en HMM, sistema basado en i-vectors y fusión de los mismos) en las sesiones de test de la base de datos Albayzin 2014

Sistema	Test
HMM	27.37 %
i-vec	22.47 %
Fusión	**20.68 %**

No obstante, a pesar de dicho aumento en el error, el 20.68 % de SER final arrojado por el algoritmo de fusión resultó suficiente para proclamar el sistema de segmentación de audio desarrollado vencedor de la campaña de evaluación Albayzin 2014, en la que, como adelantábamos en el capítulo anterior dedicado a la segmentación de audio, tomaron parte 7 sistemas desarrollados por 4 grupos de investigación diferentes.

Los resultados obtenidos en las evaluaciones de segmentación de audio 2012 y Albayzin 2014 muestran claramente el buen funcionamiento del método propuesto en este ámbito, ofreciendo una solución viable al elevado grado de desequilibrio presente en las bases de datos utilizadas, y consiguiendo una mejora realmente significativa de los resultados obtenidos por clasificadores individuales.

4.6 Extensión del método de fusión propuesto

Un aspecto importante en tareas complejas como la identificación o la verificación de locutor, es la necesidad no sólo de clasificar los distintos locutores presentes en la base de datos, sino de ofrecer además el grado de confianza en la decisión tomada. La aplicación del método de fusión propuesto en este caso requiere extender su funcionamiento a un nivel de *score*, más adecuado a este tipo de tareas.

Un primer enfoque en este caso, consiste en utilizar el valor estimado de confianza en la decisión de un clasificador, $con(k)$, a modo de *score*. Para ello, se deben calcular en primer lugar las matrices de confusión de los distintos sistemas de reconocimiento de locutor disponibles, utilizando las muestras de audio de los locutores presentes en la base de datos de entrenamiento. A continuación, se asignará a cada nueva muestra de entrada una de las clases o locutores objetivo (aquellos presentes en el entrenamiento) y se evaluará la confianza en la decisión tomada por cada uno de los sistemas disponibles.

Supongamos ahora que la nueva muestra analizada pertenece a uno de los locutores objetivo. Es de esperar que la mayor parte de los sistemas de reconocimiento identifiquen correctamente el locutor presente en el audio, asignando la misma clase de salida a dicha muestra. Si se trata de un locutor claramente diferenciado en la fase de entrenamiento, los valores implicados en el cálculo de la confianza estarán situados en la diagonal de cada una de las matrices de confusión de los distintos sistemas, por lo que podemos esperar un elevado valor de $con(k)$. En caso de pertenecer dicha muestra a un locutor peor caracterizado en el entrenamiento, es razonable pensar que los errores cometidos por los distintos sistemas en el test estarán igualmente recogidos en las matrices de confusión, por lo que a pesar de utilizar en este caso valores fuera de la diagonal en el cálculo, el valor de confianza obtenido debería ser igualmente elevado, aunque lógicamente en menor medida que el de un locutor fácilmente diferenciable.

Tomemos ahora como ejemplo una muestra de un nuevo locutor, no presente en la fase de entrenamiento. Es esperable un desacuerdo general entre los distintos sistemas de reconocimiento, que no disponen de un modelo adecuado para asignar a esta nueva muestra, por lo que los valores implicados en el cálculo de confianza en este caso estarán situados fuera de la diagonal. Es razonable pensar que los errores

cometidos (si se asigna un locutor conocido a una muestra de un nuevo locutor hablaremos igualmente de un error) por los distintos sistemas en el test no estarán en este caso recogidos en las matrices de confusión, ya que al tratarse de un nuevo locutor, cada sistema asignará la nueva muestra al modelo que mejor se ajuste en cada caso, por lo que podemos esperar un valor inferior de $con(k)$.

Como se ha comentado en la sección 4.3, los ceros de la matriz de confusión deben ser sustituidos por un valor mínimo para evitar que varios sistemas acumulen continuamente el mismo valor de confianza. Resulta imprescindible en este caso una configuración adecuada de dicho valor en la parte de entrenamiento de la base de datos, de modo que al analizar una nueva muestra sea posible diferenciar un locutor mal caracterizado de uno nuevo.

Otro problema a tener en cuenta deriva de la incapacidad del método propuesto para evaluar la confianza en una clase que no haya sido emitida por alguno de los clasificadores implicados en la fusión. En ciertas tareas como la verificación o detección de locutor, se pide a cada sistema que establezca con un determinado grado de confianza si un locutor concreto está presente en un fragmento de voz, por lo que en los casos en los que ninguno de los clasificadores asigne dicho locutor a la muestra analizada resulta imposible obtener un valor de confianza.

La solución propuesta pasa, por tanto, por definir un nuevo valor de confianza en función de la clase analizada (el locutor propuesto) y la salida emitida por los distintos clasificadores o sistemas involucrados en el proceso de fusión dado por:

$$con(m) = \prod_{k=1}^{K} P(m, l_k), \quad m = 1 \cdots M \tag{4.28}$$

$$\text{con } P(m, l_k) = \begin{cases} \frac{n^{(k)}_{ml_k}}{\sum_{i=1}^{M} n^{(k)}_{il_k}}, \text{ si } l_k = m \\ \\ \frac{n^{(k)}_{ml_k}}{\sum_{j=1}^{M} n^{(k)}_{mj}}, \text{ si } l_k \neq m \end{cases} \tag{4.29}$$

En los casos en los que la salida asignada, l_k, y el locutor propuesto para la evaluación, m, coinciden, la probabilidad $P(m, l_k)$ es estimada por medio de (4.6) para cada uno de los distintos sistemas de reconocimiento de locutor. En los casos

en los que la salida asignada y el locutor propuesto para la evaluación no coinciden y asumiendo m como la clase a la que pertenece la nueva muestra analizada, evaluamos la probabilidad $P(m, l_k)$ como se ha definido en (4.26).

Dicho de otro modo, a diferencia de lo que sucedía en el caso del algoritmo original, no disponemos de información que indique la presencia del locutor propuesto en la muestra a analizar (se trata precisamente de evaluar dicha posibilidad), salvo que alguno de los sistemas seleccione dicha clase a su salida. En el caso de que un sistema concreto asigne la muestra al locutor propuesto, la probabilidad $P(m, l_k)$ es estimada por medio de la Precisión del sistema para dicho locutor. Si por el contrario el sistema en cuestión no ha seleccionado al locutor propuesto, la probabilidad $P(m, l_k)$ viene dada por la proporción de muestras de la clase m (muestras del locutor m) etiquetadas como l_k por el sistema de reconocimiento de locutor correspondiente en la parte de entrenamiento de la base de datos.

4.7 Validación del método extendido

Una vez desarrollada la extensión del método de fusión propuesto se han realizado dos experimentos diferentes para validar la solución propuesta. En este caso se pretende evaluar el funcionamiento del algoritmo desarrollado en un campo a priori más complejo como es el reconocimiento de locutor. Múltiples métodos de fusión se han utilizado en este área para mejorar el comportamiento de sistemas de reconocimiento individuales [17] [45] [42], sin embargo, el elevado número de clases (locutores) presente en este tipo de sistemas supone un reto importante para un algoritmo de fusión de etiquetas como el propuesto.

En el primer caso se han utilizado las etiquetas proporcionadas por doce sistemas junto con las bases de datos proporcionadas en la evaluación de reconocimiento de locutor 2008 organizada por el NIST [99]. En el segundo, el método propuesto ha sido aplicado a cinco sistemas de reconocimiento de locutor utilizando para ello las bases de datos proporcionadas con motivo del i-vector challenge 2014 del propio NIST. En ambos experimentos se han utilizado las métricas propuestas en las campañas de evaluación. A continuación se muestran los resultados obtenidos.

4.7.1 NIST SRE 2008

En los últimos años el NIST ha organizado diferentes evaluaciones con el objetivo de impulsar el desarrollo de las tecnologías del habla. Algunos ejemplos representativos son las NIST SRE (Speaker Recognition Evaluation), las NIST LRE (Language Recognition Evaluation) o las NIST RT (Rich Transcription Evaluation).

Este tipo de evaluaciones establecen un marco para la evaluación de distintos sistemas de múltiples laboratorios en todo el mundo mediante unas bases de datos y unas métricas comunes para todos los participantes.

Concretamente las evaluaciones de reconocimiento de locutor, NIST SRE, que definen como principal tarea la detección de locutor, es decir, determinar si un determinado hablante está presente o no en un segmento de audio concreto, han sido seleccionadas como un marco apropiado donde llevar a cabo la evaluación del método propuesto, en su versión a nivel de *score*.

Para llevar a cabo este experimento, se han utilizado los *scores* de 12 sistemas de reconocimiento de locutor diferentes desarrollados por el laboratorio Speech

and Image Processing Unit (SIPU) de la University of Estern Finland, con motivo de su participación en la evaluación NIST SRE 2008 [99]. Más concretamente, se han utilizado los *scores* obtenidos por dichos sistemas en distintos *trials* (como se define cada segmento de prueba a evaluar) de la base de datos proporcionada por la organización de la evaluación, así como los de nuevos *trials* complementarios facilitadas en una sesión de continuación (*FOLLOWUP*) posterior a la evaluación inicial. Estos *scores* han sido traducidos a etiquetas identificando en cada caso el modelo con mayor puntuación para cada uno de los sistemas.

Una vez obtenidas las etiquetas asignadas por los sistemas a cada uno de los *trials*, se ha realizado una validación cruzada con 5 capas con el fin de evaluar el funcionamiento del algoritmo de fusión propuesto. De esta forma conseguimos evaluar todos los *trials* disponibles, realizando en cada iteración el cálculo de las matrices de confusión con 4 de las 5 capas y evaluando el rendimiento del método de fusión propuesto en la restante.

Adicionalmente, se ha utilizado la primera capa para llevar a cabo la calibración del valor mínimo permitido en las matrices de confusión y que actuará como factor de penalización para aquellos casos no representados en el entrenamiento. Nos referiremos a esta capa como la parte de desarrollo, mientras que las 4 capas restantes conformarán la parte de test.

La evaluación del funcionamiento del método de fusión propuesto se ha realizado en este caso por medio de las curvas DET (Detection Error Tradeoff) [92] y el EER (Equal Error Rate). Una curva DET representa la tasa de falso rechazo (porcentaje de muestras que pertenecen al locutor objetivo y no han sido identificadas por el sistema) en función de la tasa de falsa aceptación (porcentaje de muestras identificadas por el sistema como locutor objetivo que no pertenecen al mismo), en una escala logarítmica que aproxima el resultado a una función lineal y facilita la comparación de los distintos sistemas.

El EER es el punto de la curva en el que coinciden los valores de falso rechazo y falsa aceptación, y que determina el punto de funcionamiento equilibrado del sistema. A medida que disminuye el valor de EER, mejor se considera el sistema. En este caso, el valor de EER se ha utilizado además para llevar a cabo la calibración del factor de penalización necesario a la hora de aplicar el método propuesto.

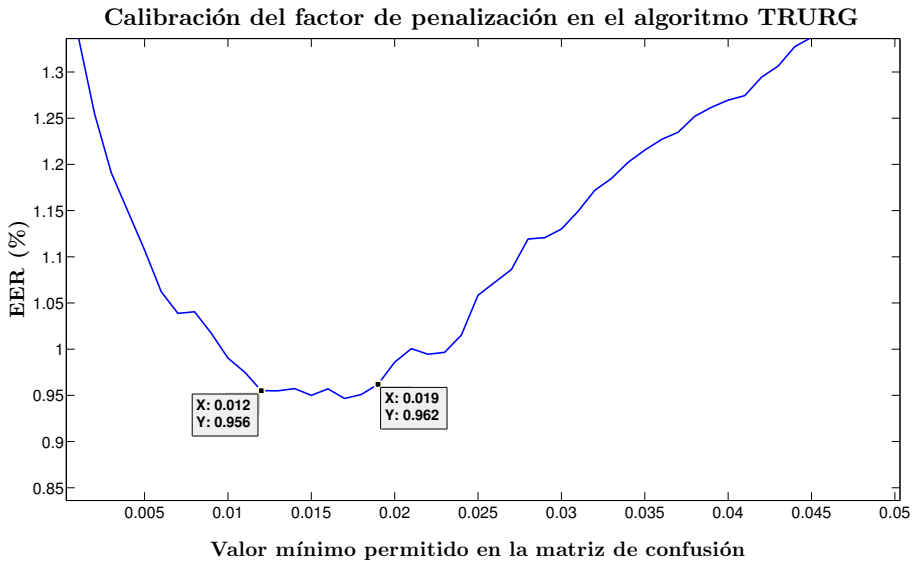


Figura 4.3: Calibración del factor de penalización en el algoritmo de fusión propuesto

La figura 4.3 muestra la evolución del EER a medida que se modifica el factor de penalización introducido en el algoritmo. Se puede observar cómo los valores comprendidos entre 0.012 y 0.019 establecen una ventana de mínimo EER, por lo que finalmente se ha fijado un factor de penalización de 0.017 que arroja un EER de 0.96 en la capa de desarrollo.

Una vez fijado el factor de penalización se han obtenido las curvas DET tanto de la parte de desarrollo como de la parte de test. La figura 4.4 muestra los resultados obtenidos por los 12 sistemas de reconocimiento de locutor desarrollados por el grupo SIPU y el obtenido al aplicar el algoritmo propuesto en las partes de desarrollo y test respectivamente.

Se puede observar cómo el método propuesto supera ampliamente la respuesta de los sistemas originales en la partes de desarrollo y de test, lo que demuestra la validez del algoritmo en entornos complejos con un elevado número de clases.

Vemos cómo los resultados obtenidos por los diferentes sistemas son similares (a excepción del sistema 8, con un error significativamente mayor al resto), por lo que podemos esperar comportamientos y decisiones muy próximas para cada nueva muestra de entrada. Sin embargo, el método de fusión propuesto es capaz de

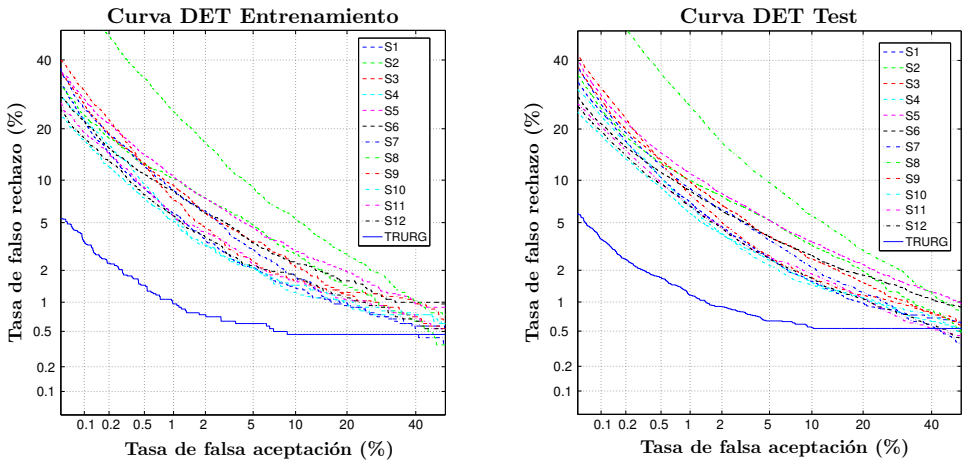


Figura 4.4: Curvas DET de los sistemas de reconocimiento de locutor y el método propuesto en la parte de desarrollo (a) y en la parte de test (b)

combinar de forma precisa las salidas de los sistemas y refinar el *score* asignado al locutor objetivo en los distintos *trials*.

Podemos observar con mayor detalle la mejora obtenida al aplicar el algoritmo de fusión de etiquetas en la tabla 4.25, donde se recogen los valores de EER de los 12 sistemas de reconocimiento de locutor y el método propuesto tanto en la parte de desarrollo como en la parte de test.

De nuevo en este caso resulta evidente la mejora obtenida por medio del algoritmo desarrollado, que obtiene un EER en torno al 1 % en ambos casos, mejorando significativamente el resultado obtenido por el mejor de los sistemas de reconocimiento actuando de forma independiente.

Tanto la tabla 4.25 como la figura 4.4 muestran cómo la combinación de decisiones permite mejorar considerablemente el funcionamiento individual de los distintos sistemas de reconocimiento. Los buenos resultados obtenidos en la parte de test indican una elevada capacidad de generalización del método de fusión de etiquetas propuesto.

Tabla 4.25: EER obtenido por cada sistema de manera individual y al aplicar la técnica de fusión propuesta a las bases de datos de desarrollo y test utilizadas

Sistema	EER Desarrollo (%)	EER Test (%)
S1	2.86	3.36
S2	4.82	5.10
S3	4.01	4.23
S4	2.72	3.13
S5	4.68	5.11
S6	4.01	4.19
S7	3.67	4.11
S8	6.79	7.32
S9	3.11	3.41
S10	2.68	3.02
S11	3.22	3.24
S12	2.91	3.26
Fusión	**0.96**	**1.11**

4.7.2 NIST i-vector challenge 2014

Para llevar a cabo este nuevo experimento se han utilizado los datos proporcionados en la consecución del NIST i-vector challenge 2014. El objetivo de este challenge organizado por el NIST conserva la tarea principal (así como las bases de datos) de las ediciones previas de las evaluaciones NIST SRE, la detección de locutor.

A diferencia de lo que ocurre en las evaluaciones NIST SRE, los segmentos de voz a evaluar se proporcionan en este caso en forma de i-vector. De esta forma se amplía el ámbito de trabajo, más orientado tradicionalmente al procesado de señal de voz, hacia el reconocimiento de patrones o el machine learning en general.

Para llevar a cabo la evaluación del comportamiento de los distintos sistemas participantes en el challenge se utiliza una función de coste de decisión (DCF), que representa una combinación lineal de las probabilidades de falso rechazo y falsa aceptación en función de un umbral.

La función de coste propuesta en este caso queda recogida en la ecuación:

$$DCF(t) = (miss(t)/target_T) + (100 \cdot falarm(t)/nontarget_T) \qquad (4.30)$$

donde $miss$ y $falarm$ representan respectivamente el número de falsos rechazos y falsas aceptaciones cometidos por el sistema. $target_T$ expresa el número de *trials* en los que el locutor a evaluar está presente en el segmento de audio, mientras que $nontarget_T$ denota el número de *trials* en los que dicho locutor no aparece en el segmento de audio que se está analizando.

El mínimo valor de DCF obtenido en función de todos los posibles umbrales se proporciona como *score* oficial del sistema, de modo que cada participante en el challenge quedará organizado por medio de este valor mínimo de DCF obtenido sobre el conjunto de *trials* que componen la parte de test de la base de datos.

Se han utilizado en este caso las salidas de 5 sistemas de reconocimiento de locutor diferentes desarrollados por el laboratorio Aholab con motivo de su participación en el challenge, todos ellos con un valor de DCF superior (y por tanto peor comportamiento) al del sistema baseline proporcionado por la organización.

Al igual que en el resto de experimentos realizados se han calculado las matrices de confusión de los 5 sistemas en la parte de entrenamiento de la base de datos, en este caso i-vectors. A continuación, se ha obtenido la salida proporcionada por cada sistema a cada nueva muestra analizada y se ha calculado el valor de confianza para el locutor evaluado en cada caso.

Además del conjunto de *trials* que forman la parte de entrenamiento, se proporciona un set de desarrollo para comprobar el funcionamiento de los métodos propuestos por los participantes del challenge. Finalmente se proporciona un set de test con nuevos *trials* que conforman la parte de evaluación final.

La tabla 4.26 recoge los resultados del sistema baseline proporcionado por la organización del NIST i-vector challenge y los obtenidos al aplicar el método de fusión de etiquetas propuesto a los 5 sistemas desarrollados. Se muestran los resultados obtenidos tanto en el set de desarrollo como en el de test.

Se puede observar cómo el método propuesto consigue nuevamente mejorar los resultados del sistema baseline a pesar de partir de las salidas de sistemas de reconocimiento de locutor con peor comportamiento. De nuevo comprobamos la validez del algoritmo en este tipo de entornos con un número elevado de clases.

Tabla 4.26: Valores DCF mínimos obtenidos por el sistema baseline propuesto por la organización y al aplicar el método propuesto a los datos del NIST i-vector challenge

Sistema	minDCF Desarrollo	minDCF Test
Baseline	0.386	0.378
Trurg	**0.354**	**0.356**

Por último, cabe destacar que aun habiendo demostrado el buen funcionamiento del método propuesto en ambos experimentos, la información acerca de los locutores no objetivo utilizada para calcular la confianza en cada una de los *trials* no está permitida en las normas que rigen las evaluaciones, por lo que ninguno de los sistemas resultantes de la fusión podría ser considerado para tomar parte en los challenges descritos. No obstante, el objetivo de los experimentos no era otro que el de evaluar el comportamiento del algoritmo en entornos complejos con un elevado número de clases, lo que se ha llevado a cabo con éxito considerable.

4.8 Conclusiones

Se han analizado en este capítulo los distintos aspectos involucrados en la tarea de fusión de clasificadores, con especial atención a los métodos que hacen uso exclusivamente de las etiquetas proporcionadas por dichos clasificadores, ya que este tipo de fusión puede suponer una buena alternativa a la hora aprovechar el conocimiento previo de los sistemas de diarización acerca de los locutores presentes en el audio, sin renunciar por ello al funcionamiento online de los mismos.

En primer lugar, se ha presentado el problema de clasificación de patrones y se han descrito los métodos de fusión de etiquetas utilizados en la literatura: Majority voting, Bayes Belief Integration, Pairwise Fusion Matrix y Behaviour-Knowledge Space. A continuación, se ha planteado el problema del comportamiento de los distintos métodos de fusión descritos al enfrentarse a un desequilibrio en la base de datos y se ha realizado un análisis detallado del funcionamiento interno de cada uno de ellos. Así mismo, se han recogido las diversas medidas presentes en la literatura que tienen en cuenta el desequilibrio de muestras entre clases.

A continuación, se ha introducido la técnica de fusión de etiquetas propuesta, diseñada para combatir el problema de desequilibrio de las bases de datos, frecuente en problemas del área del procesado de señal de voz. El nuevo método de fusión propuesto, denominado TRURG, propone estimar la confianza en la decisión de un clasificador, utilizando para ello las matrices de confusión de todos los clasificadores implicados en el proceso de fusión.

Posteriormente, se han realizado múltiples experimentos para validar el funcionamiento de la técnica de fusión propuesta en condiciones de desequilibrio mediante la utilización de distintas bases de datos del repositorio UCI y pertenecientes al área del reconocimiento de emociones. En estos experimentos, el método propuesto ha sido comparado con los distintos métodos de fusión descritos anteriormente.

En primer lugar, se han utilizado las etiquetas proporcionadas por tres clasificadores entrenados mediante el software WEKA junto con distintas bases de datos del repositorio UCI. En este caso el método TRURG, ha obtenido resultados significativamente mejores en la mayoría de bases de datos en lo referente a medidas de UAR, GMEAN, MAUC y Fscore, debido a que realiza una mejor clasificación de las muestras de las clases minoritarias en la base de datos.

A continuación se han presentado los experimentos realizados en el área del reconocimiento de emociones mediante la voz. En este caso únicamente los métodos Majority voting y TRURG han conseguido mejorar el funcionamiento de los clasificadores originales, siendo el algoritmo TRURG es el método que obtiene un mejor rendimiento en términos de UAR, Gmean, MAUC y Fscore.

Una vez comprobado el buen funcionamiento del método de fusión de etiquetas propuesto en condiciones de desequilibrio en la base de datos, se han presentado nuevos experimentos realizados en dos conocidas áreas del procesado de señal de voz: segmentación de audio y reconocimiento de locutor.

En primer lugar las campañas de evaluación Albayzin 2012 y 2014 se han utilizado como de marco de aplicación del método de fusión. En ambos casos el método propuesto ha obtenido una mejora significativa de los resultados, principalmente en la clase música, motivo de preocupación especial en la campañas Albayzin a la hora de diseñar los sistemas. Los resultados obtenidos muestran claramente el buen funcionamiento del método propuesto en el ámbito de la segmentación de audio, a pesar del elevado grado de desequilibrio presente en las bases de datos utilizadas.

Por último se han presentado los experimentos realizados en el área del reconocimiento y verificación de locutores. En primer lugar se ha presentado una extensión del método propuesto con el fin de adaptar su funcionamiento a un nivel de score, más adecuado en este tipo de sistemas. A continuación el método propuesto se ha aplicado con éxito a los datos de dos evaluaciones distintas organizadas por el NIST: NIST SRE 2008 y NIST i-vector challenge 2014. A pesar del buen funcionamiento demostrado en ambos experimentos, la información utilizada por el método propuesto supone la infracción de las normas que rigen las evaluaciones, por lo que ninguno de los sistemas resultantes de la fusión podría ser utilizado para participar en los challenges descritos.

No pinto cosas, sólo la diferencia entre las cosas.

Henri Matisse

CAPÍTULO

5

Segmentación de locutor

En el capítulo 3 hemos definido la segmentación automática como el proceso de división de un archivo de audio en secciones homogéneas de acuerdo con su contenido. La segmentación de locutor, hace referencia a la división del audio en segmentos correspondientes a los distintos locutores presentes en la grabación.

Cuando se dispone de información referente a los locutores que aparecen en el audio, es posible aplicar las técnicas basadas en modelos, descritas en el capítulo 3, para llevar a cabo la segmentación del audio. En este caso, en lugar de las distintas clases acústicas, se realiza el entrenamiento de modelos estadísticos para cada uno de los locutores objetivo. A continuación, se lleva a cabo la división del audio en segmentos que son clasificados utilizando los modelos previamente entrenados, identificando como fronteras los cambios de locutor entre segmentos consecutivos.

Sin embargo, el término de segmentación de locutores habitualmente hace referencia a las distintas técnicas que no hacen uso de ningún tipo de conocimiento previo sobre la identidad o el número de locutores presentes en el audio. El objetivo principal de la segmentación de locutores es por tanto, localizar los cambios de locutor en una grabación, por lo que a menudo nos referimos este proceso como detección de cambio de turno o simplemente detección de cambio de locutor.

En ocasiones se utiliza el término diarización para hacer referencia al proceso de segmentación de locutores. En este caso diferenciaremos la segmentación de

locutores como la detección de cambio de turno exclusivamente, utilizada como paso previo a la de agrupación de locutores en un sistema de diarización.

Para llevar a cabo la detección de cambio de turno, generalmente una medida de distancia acústica evalúa la similitud entre dos ventanas adyacentes que se van desplazando a lo largo del archivo audio, identificando los puntos de cambio o fronteras entre segmentos en base a un determinado umbral. Éste es el enfoque habitual en grabaciones monocanal.

Cuando se dispone de grabaciones de varios micrófonos es posible estimar la posición relativa de los distintos locutores presentes en la grabación, permitiendo mejorar distintos aspectos de los sistemas de diarización, y más concretamente la segmentación de locutor. Es por ello que en los últimos años, han surgido distintos métodos de segmentación basados en el análisis de las diferencias entre las señales recibidas por cada uno de los micrófonos.

Este capítulo se centra en las distintas técnicas de segmentación de locutor, basadas tanto en distancias como en análisis multicanal. En primer lugar, se recogen los métodos de detección de cambio de locutor habitualmente encontrados en la literatura. A continuación, se presentan las métricas dedicadas a la evaluación de los distintos métodos. Por último, se describen la técnica de segmentación de locutores propuesta y los resultados obtenidos en distintas bases de datos utilizadas.

5.1 Estado del arte

5.1.1 Grabación mediante micrófono distante único

En este tipo de escenario, SDM (Single Distant Microphones), la segmentación de locutor (la diarización de locutores en general), se realiza en base a la información obtenida a partir de un único canal de audio, siendo el caso más habitual el correspondiente a las señales broadcast de radio o televisión. En el ámbito de las reuniones de trabajo, donde habitualmente se realizan grabaciones con múltiples micrófonos, se recurre a la combinación de las señales disponibles en un solo canal o la utilización de la señal correspondiente al micrófono distante más centrado a la hora de realizar el proceso de segmentación.

El enfoque habitual utilizado para realizar la segmentación de locutor en este caso consiste en comparar dos ventanas adyacentes del audio mediante una métrica de distancia entre las dos, identificando si ambas ventanas han sido originadas por el mismo o distinto locutor en base a un determinado umbral.

Podemos encontrar en la literatura diferentes técnicas de segmentación basada en distancia, que difieren en la métrica, el tipo de enventanado y el umbral de decisión utilizados. Se describen a continuación los métodos de segmentación basada en distancia más ampliamente utilizados. En primer lugar se presenta además la nomenclatura necesaria para llevar a cabo la descripción de dichos métodos.

5.1.1.1 Nomenclatura

Consideremos dos segmentos de audio i y j, con sus correspondientes vectores de parámetros X_i y X_j, de longitudes N_i y N_j. Supongamos que M_i y M_j modelan los segmentos X_i y X_j. Adicionalmente, consideraremos $X = X_i \cup X_j$, de longitud N, como el vector resultante de la concatenación de X_i y X_j, modelado por M.

Del mismo modo, definimos las dos hipótesis a evaluar, H_1 y H_2, que determinarán si existe o no un cambio de locutor entre los segmentos. Denominaremos H_1 a la hipótesis nula, es decir, ambos segmentos pertenecen a un mismo locutor. Por el contrario, denominaremos H_2 a la hipótesis alternativa; existe un cambio de locutor entre los segmentos X_i y X_j.

La distancia $D(i,j)$ determinará la hipótesis correcta en cada caso. Generalmente la distancia escogida evaluará la similitud de los vectores de parámetros X_i y X_j y la homogeneidad de cada uno de estos vectores respecto a la homogeneidad de la concatenación de ambos X. Normalmente la distancia obtenida para cada una de las hipótesis se comparará con un umbral a la hora de tomar la decisión final:

$$D(i,j) \underset{H_1}{\overset{H_2}{\gtrless}} \varepsilon \tag{5.1}$$

5.1.1.2 **Criterio de Información Bayesiana**

El criterio de Información Bayesiana o BIC [128] es la métrica más ampliamente utilizada en segmentación de locutores. Se trata de un criterio probabilístico penalizado por la complejidad del modelo, utilizado normalmente para seleccionar el modelo más adecuado en cada caso. El valor de BIC que indica el nivel de ajuste del modelo M_i a un segmento dado X_i viene dado por:

$$BIC(M_i) = \log \mathcal{L}(X_i, M_i) - \lambda \frac{1}{2} \#(M_i) \log(N_i) \tag{5.2}$$

donde $\mathcal{L}(X_i, M_i)$ representa la verosimilitud obtenida por el modelo en los datos y $\#(M_i)$ el número de parámetros que definen el modelo, y siendo λ un parámetro configurable dependiente de los datos. El término $\lambda \#(M_i) \log(N_i)$ penaliza el valor de verosimilitud del modelo en función de la complejidad del mismo.

Al tratarse de un criterio probabilístico penalizado por la complejidad, generalmente se utiliza para decidir si una única distribución modela de forma más adecuada el audio enventanado que dos distribuciones diferentes, es decir, si los segmentos analizados pertenecen a un mismo locutor, o por el contrario, se ha detectado un cambio en el flujo de audio. Para determinar si existe un cambio de locutor se evalúa la hipótesis H_1, asumiendo que los datos de los segmentos $X = X_i \cup X_j$ quedan mejor representados por un único modelo M, frente a la hipótesis H_2, suponiendo M_i y M_j los modelos más adecuados, mediante la expresión:

$$\Delta BIC = BIC_{H_2} - BIC_{H_1} = R(i,j) - \lambda P \tag{5.3}$$

donde $R(i,j)$ representa la diferencia de log-verosimilitudes y P la diferencia de penalización por complejidad obtenidas para cada una de las hipótesis. Si suponemos que cada segmento se modela mediante una distribución gaussiana con matriz

de covarianza completa, $X \sim M(\mu_X, \Sigma_X)$, el término $R(i,j)$ viene dado por:

$$R(i,j) = \frac{N}{2}\log|\Sigma_X| - \frac{N_i}{2}\log|\Sigma_{X_i}| - \frac{N_j}{2}\log|\Sigma_{X_j}| \tag{5.4}$$

El término P por su parte, depende del número de parámetros del modelo, p. Suponiendo matrices de covarianzas completas, este valor viene dado por:

$$P = \frac{1}{2}(p + \frac{1}{2}p(p+1))\log(N) \tag{5.5}$$

A menudo se utilizan modelos de mezclas gaussianas para representar los datos. En el caso de utilizar GMMs para los modelos la diferencia BIC se define como:

$$\begin{aligned} \Delta BIC = &\log\mathcal{L}(X, M) \\ &- (\log\mathcal{L}(X_i, M_i) + \log\mathcal{L}(X_j, M_j)) \\ &- \lambda\Delta\#(i,j)\log(N) \end{aligned} \tag{5.6}$$

donde $\lambda\Delta\#(i,j)$ representa la diferencia entre el número de parámetros de los modelos individuales, M_i y M_j, y del modelo combinado M. El uso de esta métrica en segmentación de locutor fue propuesto por Chen y Gopalakrishnan en [26].

Un aspecto importante a tener en cuenta es el parámetro de ponderación de la penalización de complejidad, λ, que debe ser ajustado cada vez que se utilizan nuevos datos. Es por ello que se han llevado a cabo diversos estudios con el fin de obtener una configuración automática del mismo. Una de las soluciones propuestas [1], supone utilizar para el modelo combinado M el doble de parámetros que en los modelos individuales, M_i y M_j, de forma que el término de penalización $\lambda\Delta\#(i,j)$ sea cancelado, junto con la configuración del parámetro de ponderación.

La manera más habitual de implementar el algoritmo consiste en usar dos ventanas adyacentes de longitud fija que se van desplazando. La distancia BIC se calcula entre las dos ventanas y los picos de la función de distancia definen los puntos de cambio. Un esquema más elaborado que a menudo se utiliza junto a métricas BIC, hace uso de una ventana creciente. Este esquema proporciona generalmente mejores resultados, pero también es más costoso desde el punto de vista computacional. Algunas implementaciones reducen el tiempo de cálculo descartando puntos poco probables y definiendo un límite para el crecimiento de la ventana [23].

Se trata de un algoritmo complejo computacionalmente. Éste es el motivo por el que en algunos trabajos se han utilizado otras medidas, que pueden resultar más imprecisas, pero requieren un menor tiempo de cálculo.

5.1.1.3 Razón de probabilidad generalizada

La razón de probabilidad generalizada o GLR (Generalized Likelihood Ratio) evalúa la relación de verosimilitudes existente entre las hipótesis H_1 (un único modelo M representa mejor los datos de los segmentos $X = X_i \cup X_j$) y la hipótesis H_2 (los modelos individuales M_i y M_j se ajustan mejor a los datos). La relación de verosimilitudes entre las hipótesis viene dada por la expresión:

$$GLR(i,j) = \frac{H_1}{H_2} = \frac{\mathcal{L}(X,M)}{\mathcal{L}(X_i,M_i)\mathcal{L}(X_j,M_j)} \tag{5.7}$$

o en su forma de distancia:

$$D(i,j) = -\log(GLR(i,j)) \tag{5.8}$$

Este enfoque, cuya utilización para detección de cambios fue propuesto en primer lugar por Willsky y Jones en [151], difiere de la relación de verosimilitud estándar (LR) en la falta de conocimiento previo de los locutores presentes en los segmentos. En el cálculo de la GLR la función de densidad de probabilidad de los locutores es desconocida, por lo que debe ser estimada a partir de los datos contenidos en los segmentos, mientras que en el cálculo de la LR, la función de densidad de probabilidad de los locutores se proporciona como dato a priori.

5.1.2 Múltiples micrófonos distantes

A diferencia del entorno SDM, en este tipo de escenario MDM (Multiple Distant Microphones), la segmentación de locutor se realiza a partir de múltiples canales de audio, siendo el caso más habitual el de las reuniones de trabajo. En este caso, la grabación de las señales se realiza mediante micrófonos situados alrededor de los distintos locutores presentes en la habitación o en la superficie de la mesa central.

Cuando se dispone de las señales de varios micrófonos, es posible estimar el tiempo de retardo de llegada o time delay of arrival (TDOA) y por tanto, la posición relativa de los distintos locutores presentes en la grabación. En los últimos años esta técnica se ha aplicado con buenos resultados en el área de la diarización de locutores, siendo la correlación cruzada generalizada con transformación de fase o generalized cross correlation with phase transform (GCC-PHAT) es el método más comúnmente utilizado para llevar a cabo estimación del TDOA.

Uno de los enfoques más habituales consiste en utilizar la información contenida en el TDOA para obtener una señal de mejor calidad a partir de los distintos canales disponibles, siendo la técnica de retraso y suma o delay and sum beamforming el caso más habitual [145] [6]. Adicionalmente, a menudo se extraen múltiples valores TDOA (uno por cada par de micrófonos disponibles), formando un vector de características que permite mejorar el reconocimiento de los locutores presentes en el audio [104]. Así mismo, es posible combinar características acústicas (como MFCCs) y de retardos para mejorar los resultados del sistema [105].

Dado que la mayor parte de grabaciones en el entorno de reuniones de trabajo cuentan con señales multicanal, es posible utilizar la información relativa a la posición de los distintos locutores presentes en el audio para determinar la existencia de cambios de turno. En esta sección se describen los métodos de segmentación de locutor basados en MDM y TDOA encontrados en la literatura, y que sirven como punto de partida de la técnica propuesta en esta tesis.

5.1.2.1 Segmentación de locutores mediante TDOA

Tradicionalmente las técnicas de segmentación de locutor han centrado sus esfuerzos en señales de un solo canal, analizando las variaciones de las características espectrales de la señal de voz para detectar los posibles puntos de cambio de turno.

El aumento de las grabaciones multicanal, principalmente en el entorno de reuniones de trabajo, y la mejora en las técnicas de estimación de características temporales, relacionadas con el retardo entre los distintos micrófonos, han motivado la aparición de distintos métodos de segmentación basados exclusivamente en las diferencias entre las señales recibidas por cada micrófono.

En [40], la correlación cruzada entre canales se utiliza para encontrar un valor máximo que representa un valor de retardo entre dos canales. Las diferencias de tiempo se pueden estimar mediante la obtención del retardo l que maximiza la correlación cruzada centrada en un instante T:

$$\rho_{ij}[l] = \frac{\Sigma_{n=-N/2}^{N/2} m_i[n] m_j[n+l]}{\sqrt{\Sigma m_i^2 \Sigma m_j^2}} \tag{5.9}$$

donde $m_i[n]$ representa las muestras recogidas por el micrófono i y $m_j[n]$ las muestras recogidas por el micrófono j. El retardo para cada par de micrófonos $l_{i,j}$ para

un instante T viene dado por:

$$l_{ij} = \arg\max_{l} \rho_{ij}[l] \tag{5.10}$$

Una vez obtenidos los valores de retardo, se lleva a cabo un agrupamiento para identificar los segmentos de audio pertenecientes a cada uno de los locutores presentes en el audio.

Un enfoque similar se utiliza en [104], donde la segmentación de locutor se lleva a cabo utilizando únicamente los valores de retardo obtenidos mediante la correlación cruzada entre los distintos canales disponibles. Dado que no se dispone de información sobre el número de locutores y su posición relativa, resulta imposible el cálculo exacto del TDOA entre los segmentos, por lo que se propone como alternativa una versión modificada de la correlación cruzada generalizada con transformación de fase GCC-PHAT [5] definida como:

$$l(i,j) = \arg\max_{l}(R_{PHAT}(l)) \tag{5.11}$$

donde $R_{PHAT}(l)$ representa la transformada inversa de la correlación cruzada generalizada. Se elige un micrófono como referencia y se calcula el retardo de las señales procedentes del resto de micrófonos disponibles. A continuación, se forman vectores con el conjunto de valores de retardo obtenidos para llevar a cabo una segmentación y posteriores resegmentación y *clustering* para identificar los segmentos de voz pertenecientes a los distintos locutores.

Por último, en [63] se propone la extracción de una característica C_x, basada en la energía en los primeros x ms de la respuesta impulsional de la sala o Room Impulse Response (RIR), dada por:

$$C_x(\hat{h}_m) = \frac{\Sigma_{j=0}^{n_x-1}\hat{h}_m^2(j)}{\Sigma_{j=n_x}^{L_m-1}\hat{h}_m^2(j)} \tag{5.12}$$

donde n_x representa la muestra correspondiente a x ms, $\hat{h}_m$ la RIR estimada para el micrófono m y L_m la longitud del segmento en muestras. Mediante el uso de este nuevo espacio de características basadas en la RIR se consigue mejorar el resultado obtenido por un sistema clásico basado en características TDOA.

5.2 Evaluación en segmentación de locutor

Un enfoque posible a la hora de realizar el análisis de la segmentación de locutor llevada a cabo por un determinado sistema, consiste en plantear la tarea de segmentación como un problema de clasificación (o detección) en un escenario bi-clase. En este caso, el objetivo de dicha clasificación es indicar la presencia o no de un cambio de locutor en un instante concreto de la grabación.

Para ello, se comparan las marcas de tiempo correspondientes a las hipótesis de cambio de turno realizadas por el sistema de segmentación de locutores, con las marcas de referencia proporcionadas junto con las diferentes bases de datos utilizadas, obteniendo de esta forma el número de aciertos, N_A (cambios de locutor correctamente identificados), el número errores de omisión, N_O (cambios de locutor existentes no detectados), y el número errores de inserción, N_I (falsos cambios de locutor detectados), en cada caso.

En el capítulo 4 presentábamos diversas métricas disponibles en la literatura para evaluar el rendimiento de un clasificador, siendo las más ampliamente utilizadas (y sencillas) Accuracy, Recall, Precision, y Fscore. Es posible además encontrar métricas más complejas como curvas DET, curvas ROC o el área bajo la curva ROC (AUC) para llevar a cabo una comparación más exhaustiva del comportamiento de los distintos sistemas analizados.

En este trabajo se ha optado por los valores de Recall, Precisión, y Fscore para evaluar el funcionamiento de los distintos algoritmos utilizados, debido a su amplia utilización y el bajo coste necesario para el cálculo de los mismos. Para ello, es necesario reescribir las ecuaciones de la sección 4.2 para adaptarlas al problema concreto de la segmentación de locutor.

El valor de Precisión viene dado por la relación entre el número de cambios de locutor correctamente identificados y las hipótesis realizadas por el sistema:

$$P = \frac{N_A}{N_A + N_I} \tag{5.13}$$

Del mismo modo, el valor de Recall viene dado por la relación entre el número de cambios de locutor correctamente identificados y el número total de cambios de locutor presentes en las marcas de referencia:

$$R = \frac{N_A}{N_A + N_O} \tag{5.14}$$

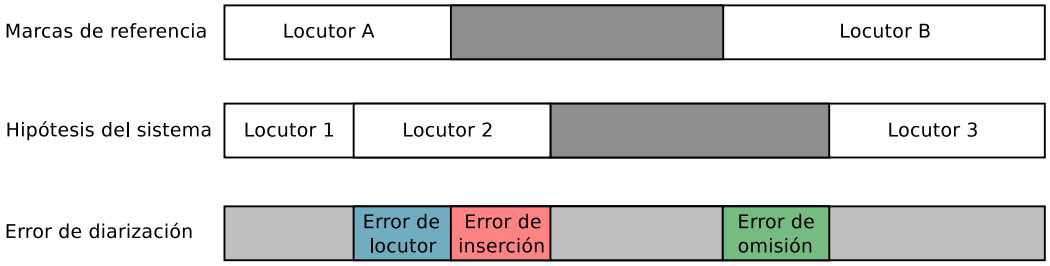


Figura 5.1: Diagrama de la descomposición del error de diarización

Una vez se han obtenido los valores de Recall y Precision, se utilizará el valor de Fscore obtenido mediante la ecuación 4.17 para llevar a cabo una mejor comparación de los distintos sistemas de segmentación de locutores.

Por otra parte, es posible analizar la segmentación de locutor realizada mediante el error de diarización final obtenido. Para ello es necesario aplicar un proceso de clustering a los segmentos originados como resultado de la etapa de segmentación. De forma paralela al SER presentado en la sección 3.2.1, el modelo de las evaluaciones organizadas por el NIST define la métrica utilizada para evaluar el funcionamiento de los sistemas de diarización de locutor como la Tasa de Error de Diarización o Diarization Error Rate (DER) [98], que se corresponde con la fracción de tiempo de locutor que no ha sido correctamente asignada.

El cálculo del error de diarización viene dado por la ecuación 3.23 identificando en este caso cada locutor como una de las clases acústicas en el caso de SER:

$$DER = \frac{\Sigma_{n\epsilon\Omega}\Xi(n)}{\Sigma_{n\epsilon\Omega}T(n)N_{ref}(n)} \tag{5.15}$$

En la figura 5.1 se muestra la descomposición del error de diarización en tres tipos de errores: error de omisión, o porcentaje de tiempo de voz de un locutor identificado por el sistema como “no voz”, error de inserción, o porcentaje de tiempo de señal asignado a un determinado locutor que corresponde a segmentos de “no voz”, y error de locutor, o porcentaje de tiempo de voz asignado a un locutor incorrecto.

5.3 Método de segmentación propuesto

Una vez presentados los distintos métodos de segmentación presentes en la literatura y las distintas métricas de evaluación utilizadas habitualmente, en esta sección se describe la técnica propuesta para la mejora de la segmentación de locutores.

Dado que las bases de datos de diarización en entorno de reuniones cuentan con grabaciones multicanal, se propone un nuevo método de segmentación que utiliza la correlación entre las señales grabadas por los distintos micrófonos para localizar posibles cambios de turno con bajo retardo de procesamiento.

Un sistema tradicional de segmentación de locutores basado en BIC proporciona hipótesis sobre posibles cambios de locutor que una posterior etapa de clustering debe reconsiderar. Este tipo de funcionamiento en dos etapas, requiere la disponibilidad de la base de datos completa antes de empezar el procesado, por lo que no es aplicable en un escenario online, donde la latencia permitida es reducida.

En este caso se propone el cálculo trama a trama de un supervector, obtenido mediante la combinación de las señales de correlación entre varias parejas de micrófonos, para identificar los posibles cambios de locutor en una grabación. La evolución del supervector entre dos tramas consecutivas determinará la existencia o no de un cambio de turno. El retardo introducido será equivalente al tamaño de ventana escogido para el cálculo de dichas correlaciones. En la figura 5.2 se muestra un diagrama con los distintos bloques que componen la etapa propuesta.

En primer lugar se realiza el enventanado de la señal. Para obtener un vector de correlaciones representativo resulta necesario utilizar ventanas de mayor tamaño

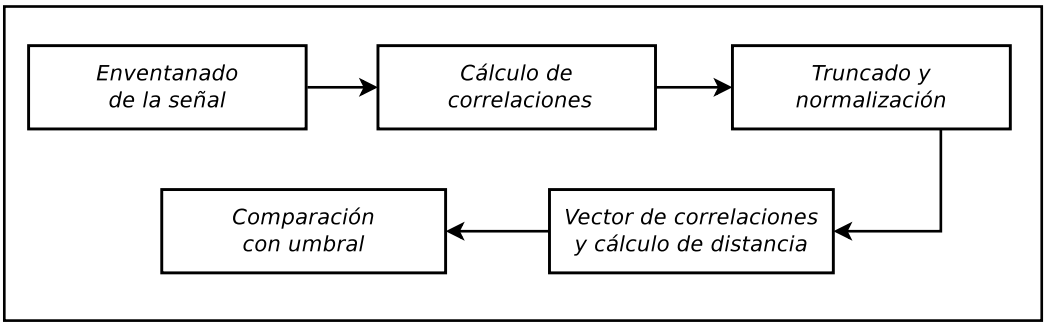


Figura 5.2: Diagrama del método de segmentación de locutores propuesto

que en el caso de características espectrales como MFCC o LPCC. Sin embargo, dado que se desea un funcionamiento online del método propuesto, adoptaremos un compromiso entre la precisión y la latencia del sistema en cada caso.

Se calculan a continuación los vectores de correlación entre los distintos micrófonos disponibles. Dado que el cálculo de las correlaciones conlleva un elevado coste computacional resulta necesaria una selección previa de los canales que van a ser incluidos en el proceso. En ocasiones la mayor parte de los pares de micrófonos proporcionan vectores de correlación similares, por lo que su utilización supondrá un aumento de redundancia en la información obtenida. Cada localización de grabación y distribución de locutores en relación con los micrófonos requerirá por tanto, de una selección de los canales que permitan llevar a cabo un proceso óptimo de segmentación de locutores.

Otro aspecto importante a tener en cuenta es el elevado número de muestras de cada vector de correlaciones. La mayor parte de la información referida a la posición relativa de locutores y micrófonos (recepción directa y primeras reflexiones) se encuentra en las muestras centrales del vector de correlación, por lo que el siguiente bloque se encarga del truncado de los vectores de correlaciones obtenido, de forma que únicamente la parte relevante de cada uno forme parte del supervector final generado. Por otra parte, resulta esperable la aparición en posiciones distintas de los valores máximos en cada vector de correlaciones (dependen directamente de las distintas posiciones de locutores y micrófonos), por lo que una vez realizado el truncado de los vectores, se lleva a cabo una normalización por máximo sobre cada uno. De esta forma se pretende conseguir un aumento en la diferenciación de los vectores obtenidos para los distintos pares de micrófonos utilizados.

Una vez se ha reducido la dimensión de los vectores de correlaciones, en el siguiente bloque se lleva a cabo la generación del vector de correlaciones global. De forma similar a la obtención del supervector-GMM, los distintos vectores de correlaciones (truncados y normalizados) correspondientes a cada par de micrófonos se concatenan en un único vector de gran dimensión que hemos denominado supervector de correlaciones. Dicho supervector, que contiene la información relativa a todos los pares de micrófonos seleccionados para el cálculo, será utilizado para determinar la existencia o no de un cambio de locutor. Para ello, en cada trama, se debe llevar a cabo una medida de distancia entre el supervector actual y el

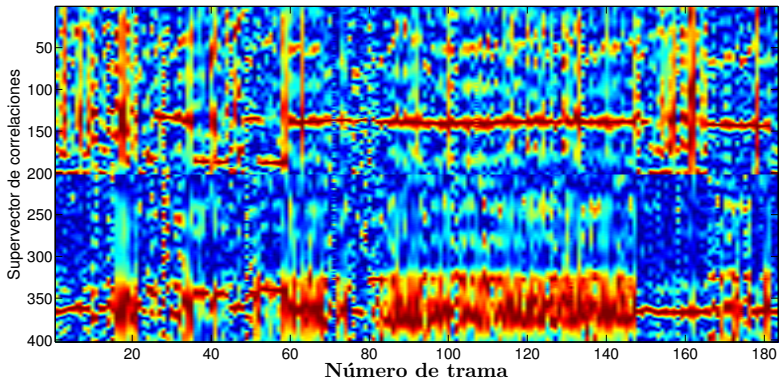


Figura 5.3: Ejemplo de la evolución trama a trama del supervector de correlaciones

correspondiente a la trama anterior. En principio, cualquier método de cálculo de la similitud entre dos vectores puede ser utilizado a la hora de tomar la decisión final, no obstante, debido a su frecuente aplicación, en este trabajo se ha optado por la distancia coseno en los experimentos realizados. En la figura 5.3 podemos observar un ejemplo de la evolución del supervector de correlaciones obtenido a partir de las señales de correlación de dos parejas de micrófonos.

Para asegurar que la información contenida en el supervector pertenece efectivamente a alguno de los locutores presentes en el audio, resulta imprescindible un correcto funcionamiento del VAD, que evite que silencios, distintos ruidos o golpes pasen al sistema de detección. En caso contrario, el supervector correspondiente a este tipo de eventos, diferente al resultante de la voz de los distintos locutores, propiciará la detección de un inexistente cambio de locutor.

El último bloque es el encargado de determinar la existencia o no de un cambio de turno. Para ello, la medida de distancia entre dos supervectores consecutivos se compara con un determinado umbral configurado en la parte de entrenamiento de la base de datos. El calibrado de dicho umbral determinará el buen funcionamiento del sistema de detección. Del mismo modo, la dimensión del supervector de correlaciones, dependiente del truncado realizado a los distintos vectores de correlación, así como el tamaño de la ventana utilizado para el cálculo de los mismos, o las distintas parejas de micrófonos a utilizar en dicho cálculo, deben ser obtenidos heurísticamente a partir de las grabaciones de entrenamiento de la base de datos.

5.4 Validación del método propuesto

Para validar la técnica de segmentación propuesta se han realizado distintos experimentos mediante dos bases de datos ampliamente utilizadas en el área de diarización de locutores, y más concretamente, en el entorno de reuniones de trabajo. En primer lugar se ha utilizado la base de datos ICSI Meetings [70], mientras que el AMI Meeting Corpus [21] ha sido utilizado en el segundo bloque de experimentos. Ambas bases de datos se han descrito en detalle en el capítulo 2.

5.4.1 Condiciones de experimentación

En ambos casos, la técnica de segmentación de locutores propuesta ha sido comparada con el método basado en distancia BIC, recogido en la sección 5.1.1.3 y ampliamente utilizado en el área segmentación de locutores. Se han utilizado para ello el sistema de diarización propuesto en [87], basado en BIC, y un segundo sistema diseñado específicamente para seguir la técnica propuesta.

Para garantizar la adecuada comparación de ambos métodos en los experimentos, se han llevado a cabo ciertas medidas en lo que a las etapas de detección de voz y de *clustering* de los sistemas de diarización se refiere.

En primer lugar, para evitar la influencia de las distintas etapas de detección de voz de los sistemas en la posterior segmentación de locutor realizada, se han utilizado las marcas de referencia de las dos bases de datos incluidas en los experimentos para realizar la segmentación voz/no voz en cada uno de los casos, simulando de esta forma el comportamiento de un detector ideal.

La grabación de las señales utilizadas en diarización en el entorno de reuniones de trabajo se realiza no obstante mediante micrófonos situados generalmente a cierta distancia de los diferentes locutores, por lo que las marcas de referencia (obtenidas a partir de los audios de micrófonos headset o de solapa) reflejan a menudo información que apenas es captada por dichos micrófonos. Los dos métodos de segmentación de locutor utilizados son sensibles al ruido, por lo que resulta esperable un aumento de los errores de inserción en ambos casos, al identificar de forma errónea estos segmentos como un instante de cambio de locutor. Especialmente perturbador resulta este efecto en el caso de la técnica propuesta, altamente sensible a silencios y ruidos como queda recogido en la sección 5.3.

Para realizar la comparación de los métodos se han utilizado las métricas descritas en la sección 5.2, es decir, número de aciertos, omisiones e inserciones, así como valores obtenidos de Precisión, Recall y Fscore, siendo este último de especial interés como reflejo del comportamiento general de cada sistema.

A continuación, para estimar la influencia de la segmentación de locutor llevada a cabo por los sistemas utilizados en el posterior error de diarización obtenido, se ha simulado un proceso de *clustering* ideal, identificando en cada segmento de audio comprendido entre dos cambios de turno consecutivos detectados, el locutor con mayor tiempo asignado en las marcas de referencia de las bases de datos.

Es posible que un proceso de *clustering* real no identifique correctamente los segmentos que pertenecen a un mismo locutor, por lo que la detección de un número excesivo de cambios de locutor puede dar lugar a problemas de sobre-*clustering* (asignar varios *clusters* distintos a un único locutor), con el consiguiente aumento del error de diarización. Resulta necesario por tanto tener este hecho en cuenta a la hora de analizar los resultados obtenidos mediante el *clustering* ideal simulado, ya que favorecerá a los sistemas de segmentación propensos a la sobre detección.

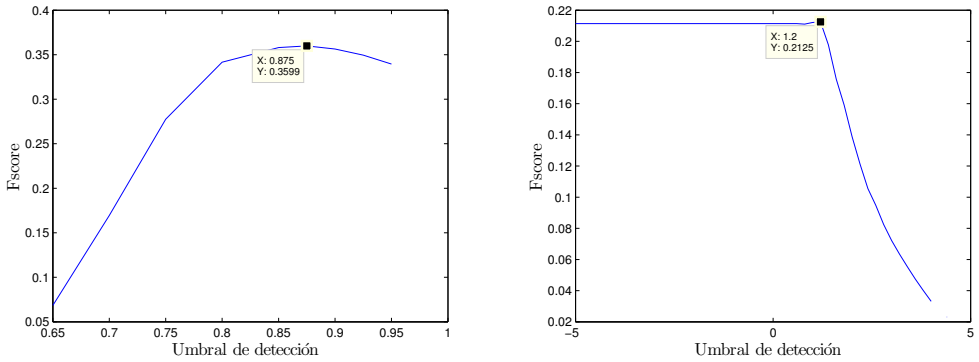
Por último, en el caso de la técnica propuesta, debe realizarse el ajuste de los parámetros de configuración (canales a utilizar, coeficientes de correlación, señales de correlación que forman el supervector...) para cada distinta localización y distribución de locutores en relación con los micrófonos. Para observar el efecto de un posible desajuste en el sistema con motivo de un cambio de localización o una nueva distribución de los locutores, únicamente la primera señal de cada una de las bases de datos ha sido utilizada para llevar a cabo la optimización de parámetros. En cuanto a los umbrales de detección, en ambos sistemas se ha realizado un ajuste, utilizando en cada caso el valor que mejores resultados ha proporcionado a lo largo del set de entrenamiento designado en cada base de datos.

Para el cálculo de las correlaciones se ha seleccionado un tamaño de ventana de 640 ms. De esta forma se establece un compromiso entre el funcionamiento online del algoritmo y la fiabilidad en el cálculo de las correlaciones.

5.4.2 Experimentos con la base de datos ICSI Meetings

En esta sección se describen los experimentos realizados utilizando las señales de la base de datos ICSI Meetings. Se ha utilizado en concreto un conjunto de 75 sesiones repartidas en dos subsets diferentes. Las 25 primeras sesiones han definido el grupo de entrenamiento, utilizado para realizar el calibrado de los sistemas de detección de cambio de locutor. Las 50 sesiones restantes han sido reservadas para llevar a cabo la evaluación de los sistemas. Se han utilizado las señales de los micrófonos PZM distribuidos uniformemente sobre las mesas centrales.

En primer lugar se ha llevado a cabo el calibrado de los sistemas, identificando en cada caso el umbral de detección que presenta un mejor resultado general, por lo que la métrica seleccionada ha sido el valor de Fscore. La figura 5.4 muestra los valores de Fscore obtenidos por ambos sistemas en el proceso de optimización de los umbrales de detección sobre las señales de entrenamiento. El umbral de detección que proporciona mayor valor de Fscore ha sido seleccionado en cada caso. Por último, se ha utilizado la primera señal de la base de datos para realizar empíricamente el ajuste de la dimensión del supervector de correlaciones, y de las señales de correlación que forman el supervector, para lo que se han utilizado en este caso dos pares de micrófonos.



(a) Optimización sistema correlaciones **(b)** Optimización sistema BIC

Figura 5.4: Optimización del umbral de detección en el sistema (a) basado en supervector de correlaciones y (b) basado en BIC en función del Fscore obtenido en la parte de entrenamiento de la base de datos ICSI Meetings

Tabla 5.1: Resultados obtenidos por ambos métodos en las sesiones de entrenamiento de la base de datos ICSI Meeting en términos de Recall, Precision, Fscore y DER

	Supervector-correlaciones				BIC			
Sesión	R.	P.	Fsc.	DER	R.	P.	Fsc.	DER
1-25	**0.641**	0.264	**0.360**	**19.28**	0.161	**0.329**	0.213	21.50

La Tabla 5.1 muestra los resultados obtenidos en el subconjunto de entrenamiento de la base de datos ICSI meetings. Se muestran los valores de Recall, Precision, Fscore y DER obtenidos mediante la aplicación de ambas técnicas de segmentación, (supervector de correlaciones y métrica BIC). Para evitar el exceso de datos proporcionados se muestran los valores medios para las 25 señales de entrenamiento. Se muestran en negrita los mejores valores obtenidos en cada caso.

Podemos observar cómo el método propuesto presenta mejores resultados respecto al sistema BIC en cuanto a valores de Recall, Fscore y DER se refiere. El método basado en supervectores de correlaciones detecta correctamente mayor número de cambios locutor, por lo que consigue un elevado valor de Recall, 0.641, mientras que el valor de Fscore se ve penalizado por un menor valor de Precision, debido a los errores de inserción cometidos por el sistema. Estos resultados muestran claramente la sensibilidad del sistema al correcto funcionamiento del VAD, implementado idealmente en este experimento mediante las marcas de referencia de la base de datos. Los silencios y ruidos pasan a la etapa de detección y generan la detección de un cambio de locutor que no se ha producido realmente. La aplicación de un VAD más adecuado supondría una mejora notable de los resultados. No obstante, la técnica propuesta presenta los mejores resultados de Fscore y DER.

El sistema basado en métrica BIC detecta menor número de cambios de locutor, por lo que obtiene valores inferiores de Fscore y mayores de DER. Sin embargo, presenta menor número de inserciones por lo que consigue mayor valor de Precision. No obstante el valor obtenido en este caso, 0.329, está lejos del óptimo.

En las siguientes figuras se muestran los valores de Fscore y DER obtenidos por los métodos en las sesiones de entrenamiento de la base de datos. Ambas métricas han sido seleccionadas porque representan el comportamiento global de los siste-

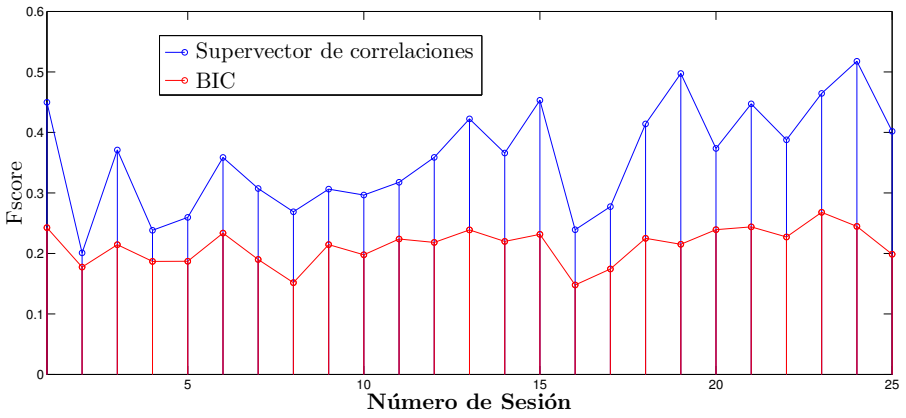


Figura 5.5: Fscore obtenido en las sesiones de entrenamiento de la base de datos ICSI

mas. Se muestran en color azul los resultados obtenidos por el método propuesto, mientras que los del sistema basado en BIC aparecen en color rojo.

En primer lugar, la figura 5.5 muestra los valores de Fscore obtenidos por ambos métodos de segmentación a lo largo de las 25 sesiones de entrenamiento. Se puede observar cómo la técnica propuesta consigue mayores valores de Fscore en todas las sesiones del subconjunto de entrenamiento. Como se ha comentado anteriormente, el método propuesto detecta mayor número de cambios locutor, por lo que consigue valores de Recall más elevados. Por el contrario, el sistema basado en métrica BIC presenta menor número de errores de inserción, por lo que consigue mayores valores de Precision. No obstante, estos valores son reducidos y el Fscore general obtenido es significativamente inferior a los de la técnica propuesta.

La figura 5.6 muestra los valores de DER obtenidos por ambos métodos a lo largo de las 25 sesiones de entrenamiento. Se puede observar cómo el método basado en BIC presenta mayores valores de DER en 24 de las 25 sesiones del subconjunto de entrenamiento, lo que demuestra el buen funcionamiento de la técnica propuesta. En este caso, sin embargo, la diferencia entre ambas técnicas es menor, con una diferencia media absoluta en torno al 2 % en las sesiones de entrenamiento.

Cabe recordar que el proceso de *clustering* implementado es ideal, por lo que la diferencia obtenida en el DER es orientativa. En ambos casos el resultado obtenido es óptimo (sin tener en cuenta el solapamiento entre locutores), sin embargo,

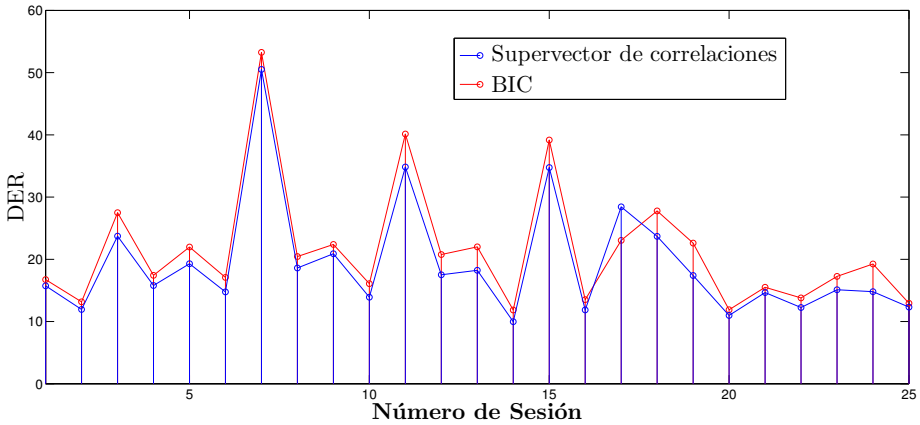


Figura 5.6: DER obtenido en las sesiones de entrenamiento de la base de datos ICSI

los cambios de locutor no detectados por el método basado en BIC darán lugar a *clusters* con audio de distintos locutores, difícilmente recuperables mediante un proceso de *clustering* realista. En el caso de la técnica propuesta, la detección de demasiados cambios de locutor puede dar lugar a problemas de sobre-*clustering* que sin embargo, pueden ser solucionados mediante un procesado posterior.

Similares resultados se han obtenido en el subconjunto de evaluación de la base de datos ICSI Meetings. Para llevar a cabo el procesado de este conjunto de señales se han utilizado nuevamente los umbrales de detección optimizados mediante las sesiones de entrenamiento de la base de datos.

La Tabla 5.2 muestra los los valores de Recall, Precision, Fscore y DER obtenidos lo largo de las 50 sesiones que forman el subconjunto. De nuevo se muestran únicamente los valores medios obtenidos para las 50 señales, utilizando la negrita para los mejores valores obtenidos en cada caso.

Tabla 5.2: Resultados obtenidos por ambos métodos en las sesiones de evaluación de la base de datos ICSI Meeting en términos de Recall, Precision, Fscore y DER

	Supervector-correlaciones				BIC			
Sesión	R.	P.	Fsc.	DER	R.	P.	Fsc.	DER
26-75	**0.687**	0.245	**0.336**	**19.37**	0.144	**0.319**	0.196	21.93

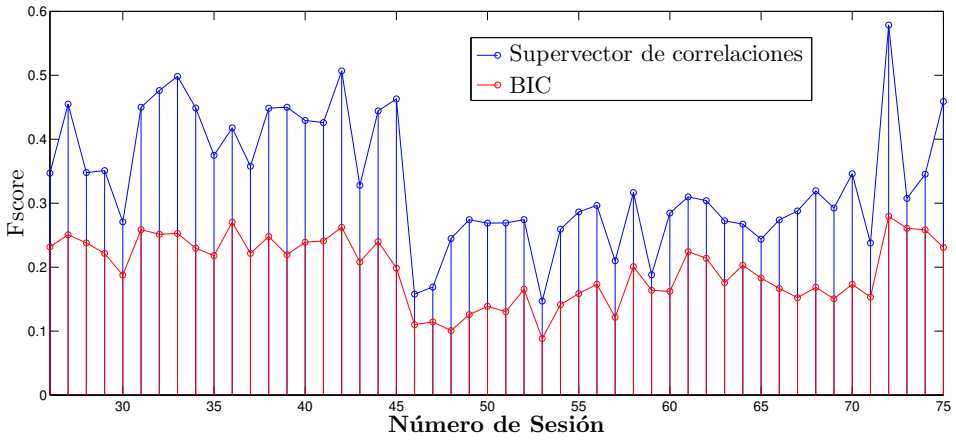


Figura 5.7: Fscore obtenido en las sesiones de evaluación de la base de datos ICSI

Podemos observar cómo nuevamente el método propuesto presenta mejores resultados respecto al sistema BIC en cuanto a valores de Recall, Fscore y DER se refiere. El método basado en correlaciones detecta correctamente mayor número de cambios locutor, consiguiendo en este caso un valor de Recall de 0.687. Los errores de inserción cometidos de nuevo penalizan los valores de Precision y Fscore, lo que demuestra la dependencia del sistema de un adecuado VAD.

La figura 5.7 muestra los valores de Fscore obtenidos por ambos métodos de segmentación en las 50 sesiones que componen el subconjunto de evaluación. Se puede observar cómo la técnica propuesta consigue nuevamente mayores valores de Fscore en todas y cada una de las sesiones evaluación.

Los resultados son similares a los obtenidos en el subconjunto de entrenamiento. La técnica propuesta detecta correctamente mayor número de cambios locutor, consiguiendo por tanto valores de Recall más elevados. El sistema basado en métrica BIC consigue mayores valores de Precision al cometer menor número de errores de inserción, aunque estos valores son reducidos y el Fscore general obtenido es significativamente inferior a los de la técnica propuesta en todas las sesiones.

Por último, la figura 5.8 muestra los valores de DER obtenidos por ambos métodos de segmentación a lo largo de las 50 sesiones del subconjunto de evaluación.

Se puede observar cómo el método basado en BIC presenta mayores valores de DER en 45 de las 50 sesiones del subconjunto de evaluación, lo que demuestra

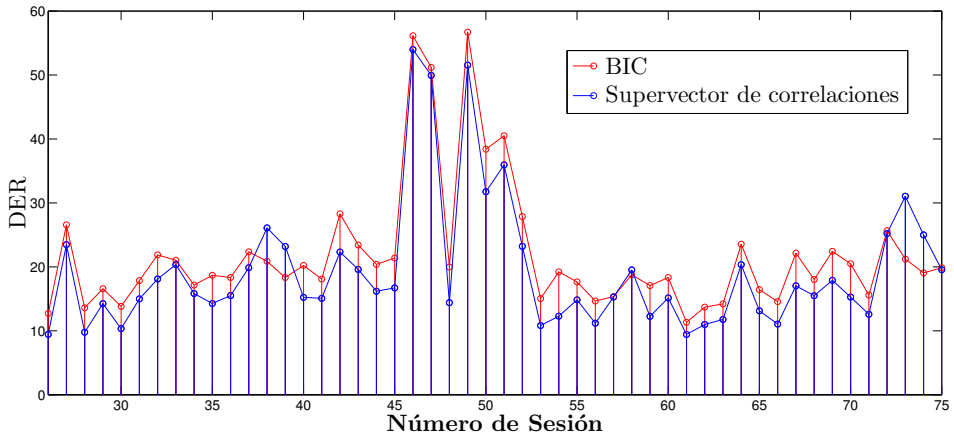


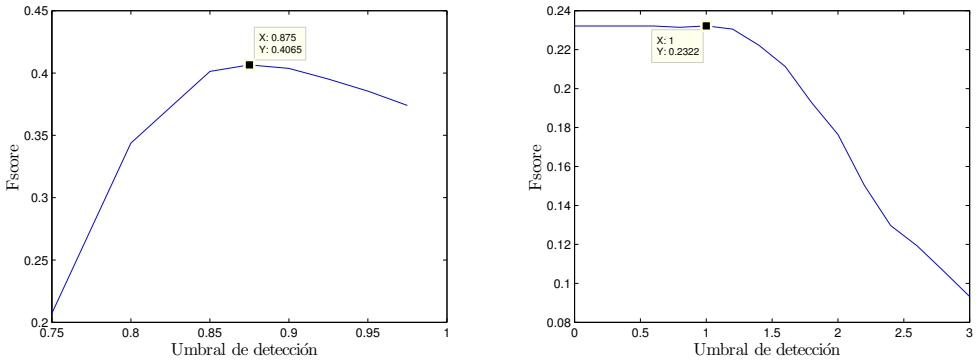
Figura 5.8: DER obtenido en las sesiones de evaluación de la base de datos ICSI

el buen funcionamiento de la técnica propuesta en el total de la base de datos. De nuevo en este caso, la diferencia entre las técnicas de segmentación en el DER obtenido es menor, con una diferencia media absoluta en torno al 2.5 % en las sesiones de evaluación. Como se ha comentado anteriormente, resulta esperable que los cambios de locutor no detectados por el método basado en BIC presenten un reto mayor a un proceso de *clustering* realista.

Estos experimentos demuestran que el método propuesto puede ser utilizado con éxito en bases de datos de reuniones de trabajo. La técnica propuesta consigue detectar correctamente mayor número de cambios locutor, consiguiendo valores de Fscore más elevados y menores valores de DER que el sistema basado BIC en la mayoría de casos. El método propuesto presenta además la ventaja de mantener el funcionamiento online de los sistemas que así lo requieran.

5.4.3 Experimentos con la base de datos AMI Meeting Corpus

En esta sección se describen los experimentos realizados utilizando las señales de la base de datos AMI Meeting Corpus, perteneciente igualmente al ámbito de las reuniones de trabajo. En este trabajo se ha utilizado un conjunto de 167 sesiones repartidas en dos subconjuntos diferentes. Para mantener la estructura del experimento anterior, las 25 primeras grabaciones han sido seleccionadas como sesiones



(a) Optimización sistema correlaciones **(b)** Optimización sistema BIC

Figura 5.9: Optimización del umbral de detección en el sistema (a) basado en supervector de correlaciones y (b) basado en BIC en función del Fscore obtenido en la parte de entrenamiento de la base de datos AMI Meeting Corpus

de entrenamiento, y utilizadas para llevar a cabo el calibrado de los sistemas de detección de locutor. Las 142 grabaciones restantes han sido reservadas para realizar la evaluación de los sistemas. Las señales utilizadas corresponden a los micrófonos de un array circular situado en la mesa central, por lo que la menor distancia entre micrófonos será un factor a tener en cuenta en este caso.

El primer paso ha sido nuevamente realizar el calibrado de los sistemas, identificando el umbral de detección que presenta mejor resultado general. La figura 5.9 muestra los valores de Fscore obtenidos por ambos sistemas en las señales de entrenamiento. Se puede observar cómo el umbral de detección óptimo en el caso de la técnica propuesta coincide con el valor obtenido para la base de datos ICSI Meetings, lo que muestra en cierta medida la portabilidad del método a diferentes bases de datos o entornos. A pesar de contar con 8 señales en el array, se ha mantenido la configuración de dos pares de micrófonos para formar el supervector de correlaciones en la implementación de la técnica propuesta, ya que los resultados obtenidos con mayor número de pares involucrados no presentan mejoras significativas.

La Tabla 5.3 muestra los resultados obtenidos en la sesiones de entrenamiento de la base de datos. Siguiendo el ejemplo del experimento anterior se muestran los valores medios de Recall, Precision, Fscore y DER obtenidos mediante la aplicación de ambas técnicas de segmentación, utilizando la negrita para los mejores

Tabla 5.3: Resultados obtenidos por ambos métodos en las sesiones de entrenamiento de la base de datos AMI Meeting en términos de Recall, Precision, Fscore y DER

	Supervector-correlaciones				BIC			
Sesión	R.	P.	Fsc.	DER	R.	P.	Fsc.	DER
1-25	**0.726**	0.288	**0.407**	**28.05**	0.181	**0.342**	0.232	32.03

valores obtenidos en cada caso.

Podemos observar cómo el resultado resulta muy similar al obtenido en el caso de la base de datos ICSI Meetings. El método propuesto presenta mejores resultados en cuanto a valores de Recall, Fscore y DER se refiere, ya que detecta correctamente mayor número de cambios locutor. Nuevamente consigue un elevado valor de Recall, 0.726, mientras que el valor de Fscore se ve penalizado por los errores de inserción cometidos por el sistema. Una vez más resulta notable la falta de un módulo de VAD más adecuado, no obstante, el valor de Fscore obtenido en este caso es significativamente superior. El sistema basado en métrica BIC presenta mayor valor de Precision ya que comete un menor número de errores de inserción, sin embargo, el valor obtenido igualmente en este caso, 0.342, está lejos del óptimo.

La figura 5.10 muestra los valores de Fscore obtenidos por ambos métodos de segmentación a lo largo de las 25 sesiones de entrenamiento.

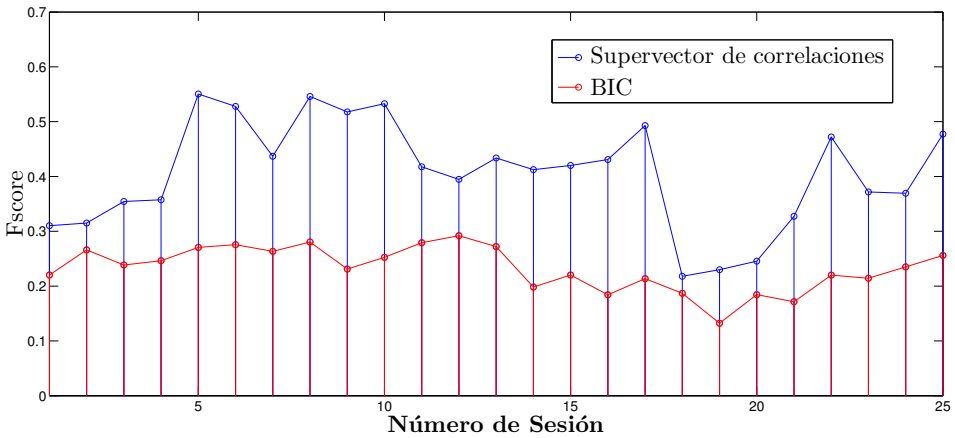


Figura 5.10: Fscore obtenido en la parte de entrenamiento de la base de datos AMI

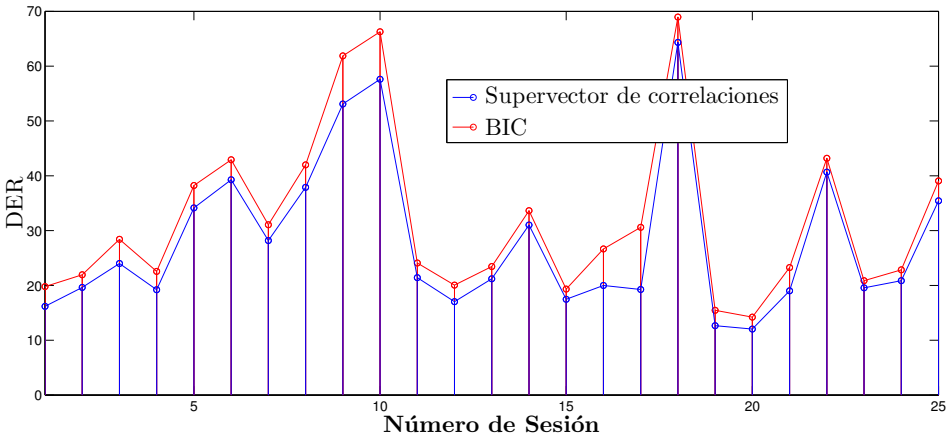


Figura 5.11: DER obtenido en las sesiones de entrenamiento de la base de datos AMI

Se puede observar cómo nuevamente la técnica propuesta obtiene mayores valores de Fscore en todas las sesiones del subconjunto de entrenamiento, consiguiendo además mayor diferencia en los valores en este caso. El método propuesto detecta correctamente mayor número de cambios locutor, por lo que consigue valores de Recall, y por tanto Fscore, significativamente superiores. Por el contrario, el sistema basado en métrica BIC presenta menor número de errores de inserción y mayores valores de Precision, aunque dichos valores son reducidos y el Fscore general obtenido es significativamente inferior a los de la técnica propuesta.

Del mismo modo, la figura 5.11 muestra los valores de DER obtenidos por ambos métodos de segmentación a lo largo de las 25 sesiones de entrenamiento.

Se puede observar cómo en este caso el método basado en BIC presenta mayores valores de DER en todas las sesiones del subconjunto de entrenamiento, lo que sin duda demuestra el buen funcionamiento de la técnica propuesta. La diferencia obtenida entre ambas técnicas en esta base de datos es mayor, con una diferencia media absoluta en torno al 4 % en las sesiones de entrenamiento. Recordemos no obstante, que el proceso de *clustering* implementado es ideal, por lo que la diferencia obtenida en el DER es orientativa.

Se muestran a continuación los resultados obtenidos en el subconjunto de evaluación de la base de datos AMI Meeting Corpus. Se han utilizado nuevamente los

Tabla 5.4: Resultados obtenidos por ambos métodos en las sesiones de evaluación de la base de datos AMI Meeting Corpus en términos de Recall, Precision, Fscore y DER

	Supervector-correlaciones				BIC			
Sesión	R.	P.	Fsc.	DER	R.	P.	Fsc.	DER
26-167	**0.737**	0.291	**0.412**	**27.40**	0.181	**0.334**	0.229	31.25

umbrales de detección optimizados mediante de las sesiones de entrenamiento de la base de datos a la hora de realizar el procesado de este conjunto de señales.

La Tabla 5.4 muestra los los valores de Recall, Precision, Fscore y DER obtenidos lo largo de las 142 grabaciones que forman el subconjunto de evaluación. De nuevo se muestran únicamente los valores medios obtenidos.

Podemos observar cómo nuevamente el método propuesto presenta valores mayores de Recall, Fscore y DER. La técnica propuesta consigue detectar correctamente mayor número de cambios locutor, consiguiendo en este caso un elevado valor de Recall de 0.737. Los errores de inserción cometidos de nuevo penalizan, aunque en menor medida en este caso, los valores de Precision y Fscore, lo que de nuevo muestra la dependencia del sistema de un adecuado VAD.

La figura 5.12 muestra los valores de Fscore obtenidos por ambos métodos de segmentación en las 142 sesiones que componen el subconjunto de evaluación.

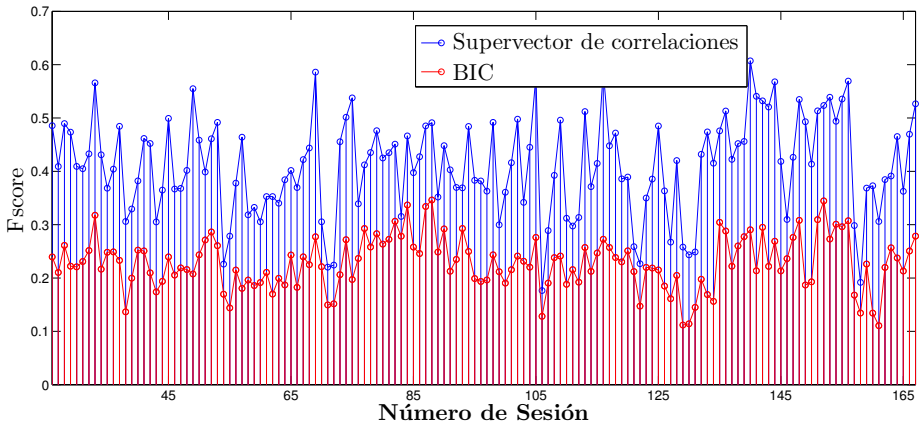


Figura 5.12: Fscore obtenido en la parte de evaluación de la base de datos AMI

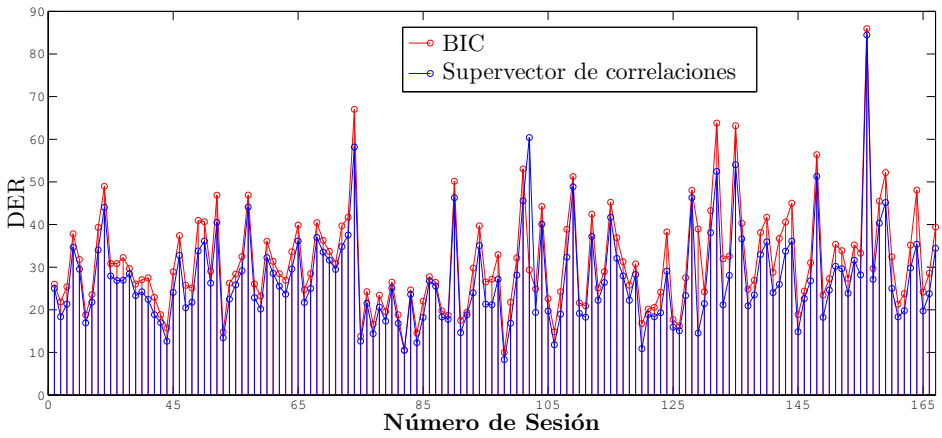


Figura 5.13: DER obtenido en las sesiones de evaluación de la base de datos AMI

Se puede observar cómo una vez más la técnica propuesta consigue mayores valores de Fscore en todas las sesiones. Los resultados son muy similares a los obtenidos en el subconjunto de entrenamiento. La técnica propuesta detecta mayor número de cambios locutor, consiguiendo elevados valores de Recall, y valores de Fscore significativamente superiores a los del sistema BIC en todas las sesiones.

Para finalizar la sección, la figura 5.13 muestra los valores de DER obtenidos por ambos métodos de segmentación en las sesiones del subconjunto de evaluación.

Se puede observar cómo el método basado en BIC presenta mayores valores de DER en las 142 sesiones del subconjunto de evaluación, lo que demuestra el buen funcionamiento de la técnica propuesta en el total de la base de datos. De nuevo en este caso, la diferencia entre las técnicas de segmentación en el DER obtenido es mayor, con una diferencia media absoluta en torno al 4 % en las 142 sesiones.

Estos experimentos, junto con los realizados con la base de datos ICSI Meetings, demuestran que el método propuesto presenta un buen funcionamiento en bases de datos de reuniones de trabajo. En este caso la técnica propuesta consigue detectar correctamente mayor número de cambios locutor, consiguiendo valores de Fscore más elevados y menores valores de DER que el sistema basado BIC en todas las sesiones de la base de datos. El funcionamiento online de la técnica propuesta se presenta además como una ventaja adicional sobre este sistema.

5.5 Conclusiones

Se han analizado en este capítulo los aspectos más importantes involucrados en la tarea de segmentación de locutor, definida como una de las etapas críticas en el proceso de diarización de locutores.

En primer lugar, se ha recurrido a la literatura en busca de diferentes técnicas de segmentación basadas en distancia, con especial énfasis en los métodos más ampliamente utilizados: Criterio de Información Bayesiana y Razón de Probabilidad Generalizada. Del mismo modo, dado que las señales multicanal predominan en el entorno de reuniones de trabajo, se han recogido los distintos métodos de segmentación de locutor basados en MDM y TDOA encontrados en la literatura. Así mismo, se han presentado las métricas dirigidas a evaluar el funcionamiento de los distintos algoritmos de segmentación, como son Recall, Precision, Fscore y Error de diarización, más ampliamente utilizadas en la literatura.

A continuación, se ha presentado una nueva técnica para la mejora de la segmentación de locutores. Se ha definido dicha técnica mediante el cálculo trama a trama de un supervector, obtenido como resultado de la concatenación de las señales de correlación de varias parejas de micrófonos, determinando la evolución de dicho supervector entre dos tramas consecutivas la existencia o no de un cambio de turno. Además de utilizar la información relativa a la posición de los distintos locutores mediante el análisis multicanal, la técnica propuesta, basada en análisis trama a trama, presenta de esta forma la ventaja de mantener el funcionamiento online de los sistemas de diarización que así lo requieran.

Se han realizado por último distintos experimentos para validar la técnica de segmentación propuesta mediante dos bases de datos ampliamente utilizadas en el entorno de reuniones de trabajo: ICSI Meetings y AMI Meeting Corpus. La técnica basada en distancia BIC, una de las más ampliamente utilizadas en la literatura, ha sido seleccionada como método de contraste en los experimentos realizados.

Los experimentos realizados mediante la aplicación de la técnica de segmentación propuesta a las señales de la base de datos ICSI Meetings han mostrado mejores resultados que los obtenidos mediante la técnica basada en distancia BIC en lo referente a las métricas de Recall, Fscore y DER, detectando correctamente mayor número de cambios locutor y consiguiendo elevados valores de Recall. Por

otro lado, se han obtenido menores valores de Precision debido principalmente a los errores de inserción cometidos, lo que muestra cierta sensibilidad del método propuesto al correcto funcionamiento del VAD.

Los experimentos realizados utilizando las señales de la base de datos AMI Meeting Corpus han mostrado resultados muy similares a los obtenido en el caso de la base de datos ICSI Meetings. El método propuesto ha obtenido mejores valores de Recall, Fscore y DER, detectando nuevamente mayor número de cambios locutor y consiguiendo elevados valores de Recall. De nuevo el valor de Fscore se ha visto penalizado por los errores de inserción cometidos, lo que muestra la importancia de un módulo de VAD adecuado a la hora de aplicar el método de segmentación propuesto. En este caso la técnica propuesta ha obtenido mayores valores de Fscore y menores valores de DER en todas y cada una de las sesiones de la base de datos utilizadas en los experimentos.

Los experimentos realizados demuestran que el método propuesto presenta un buen funcionamiento en bases de datos de reuniones de trabajo, detectando correctamente mayor número de cambios locutor y consiguiendo valores de Fscore más elevados y menores valores de DER que el método basado en distancia BIC. El funcionamiento online de la técnica propuesta se presenta además como una ventaja adicional sobre los sistemas tradicionales basados en BIC.

Tenga cuidado con las cosas pequeñas. Su ausencia o presencia pueden cambiarlo todo.

Han Shan

CAPÍTULO

6

Postprocesado de marcas

En el capítulo de introducción veíamos cómo los sistemas de diarización incorporan a menudo una etapa de resegmentación. El objetivo de esta etapa es refinar las fronteras establecidas tras la de segmentación de locutores, utilizando para ello modelos entrenados para cada uno de los locutores encontrados, a partir de los segmentos asignados por la etapa de agrupamiento de locutores a cada uno de ellos.

Dado que ahora se dispone de información suficiente, es posible entrenar modelos para cada locutor y llevar a cabo una nueva segmentación junto con modelos de no-voz, mejorando de esta forma los resultados obtenidos previamente. Se trata de un proceso que se puede repetir iterativamente, ya que el resultado de la nueva segmentación, más precisa, permite entrenar mejores modelos de locutor (más puros), que a su vez pueden ser utilizados en una nueva segmentación que mejore los resultados de la primera iteración.

No obstante, este enfoque está limitado en la práctica por el número de locutores y la cantidad de audio disponible para entrenar los modelos correspondientes. En entornos con pocos locutores, con largas intervenciones por parte de cada locutor, es posible mejorar los resultados obtenidos por el sistema de diarización básico por medio de la resegmentación del audio. Por el contrario, en entornos con muchos locutores, con intervenciones de menor duración (programas de noticias broadcast,

reuniones de trabajo...) la falta de material perteneciente a cada uno de los locutores presentes en las grabaciones, deriva en modelos poco robustos y posteriores procesos de resegmentación de menor precisión que, en la mayoría de los casos, presentan resultados inferiores a los obtenidos por el sistema básico.

En estos casos, la etapa de resegmentación a menudo es sustituida por uno o más bloques de postprocesado, generalmente de bajo impacto computacional, destinados a solucionar problemas concretos de las distintas etapas del sistema de diarización. Algunas de las técnicas utilizadas habitualmente incluyen mejora de la segmentación voz no-voz [8], selección de segmentos para la creación de modelos más robustos [13], detección de solapamiento de locutores [66], o análisis de segmentos cortos para ajuste de fronteras [55]. Otras técnicas más complejas persiguen la fusión de la salida de dos sistemas de diarización diferentes para mejorar los resultados obtenidos por cada uno de ellos individualmente [143].

Este capítulo se centra en las técnicas de mejora de la diarización basadas en el postprocesado y análisis de la información proporcionada por un sistema de diarización base. En primer lugar, se recogen los distintos métodos de mejora encontrados en la literatura. A continuación, se describen las técnicas de mejora propuestas y los resultados obtenidos al aplicar dichas mejoras a distintos sistemas de diarización sobre diferentes bases de datos.

6.1 Estado del arte

6.1.1 Mejora de la segmentación voz no-voz

La sección 5.2 presenta el error de diarización (DER) como la suma tres elementos: error de clase, error de omisión, y error de inserción. Tanto errores de omisión como de inserción son el resultado de un funcionamiento inadecuado de la etapa de detección de voz. Segmentos de música, silencio o ruido, que no son identificados en esta primera etapa, se entregan a los módulos de segmentación y de agrupación de locutores y generalmente reciben una etiqueta de locutor, incrementando el error de inserción, y en consecuencia, el error de diarización global.

En este punto, la identificación de estos segmentos y su posterior reetiquetado como "no-voz" supone por tanto una reducción del error de diarización obtenido por el sistema. Adicionalmente, permite un mejor modelado de los locutores presentes en la grabación, al eliminar los segmentos de música o ruido del proceso de entrenamiento de los modelos, mejorando así el rendimiento de una posible etapa de resegmentación posterior.

Un ejemplo de este enfoque lo encontramos en [8] [162] [9], donde se presenta un bloque de post-procesado que permite refinar las fronteras obtenidas por un sistema de diarización mediante las transcripciones proporcionadas por un sistema de reconocimiento automático del habla. Se identifican los segmentos de silencio de corta duración que no han sido eliminados por la etapa de detección de voz y se etiquetan como "no voz" las pausas inter-palabra con duración superior a un segundo. Este bloque de postprocesado supone una mejora del 0.6 % en el error de diarización, debido principalmente a la reducción de errores de inserción cometidos en la etapa de detección de voz.

6.1.2 Selección de segmentos

La etapa de resegmentación se basa en el entrenamiento de modelos a partir de los segmentos asignados por el sistema de diarización base a cada uno de los locutores presentes en el audio. Sin embargo, la segmentación llevada a cabo por este tipo de sistemas puede no ser suficientemente precisa en algunos casos, de forma que determinados clusters contienen a menudo información de locutores distintos.

Este hecho da lugar a modelos que no representan correctamente a los locutores objetivo, y resultará en un proceso de resegmentación desfavorable.

Una técnica que permite la mejora de los resultados del sistema, sin modificar para ello su funcionamiento o estructura, consiste en realizar una selección de los segmentos "puros" (que contienen audio perteneciente al locutor objetivo exclusivamente) a la hora de llevar a cabo el entrenamiento de los modelos. De esta forma, se obtienen modelos de locutor más robustos, que posteriormente darán lugar a un proceso de resegmentación más preciso y una reducción del error de diarización.

En [13], un algoritmo de selección basado en voto por mayoría y verosimilitud máxima normalizada proporciona scores para evaluar la pureza de los segmentos encontrados en el audio. Mediante la aplicación de esta técnica de selección de segmentos se presenta una mejora relativa del 29 % en el DER respecto al sistema de diarización base.

6.1.3 Detección de solapamiento de locutores

Tradicionalmente los sistemas de diarización convencionales proporcionan una etiqueta única a los segmentos de audio (por medio de una segmentación mediante algoritmo de Viterbi por ejemplo), por lo que el solapamiento supone una fuente considerable del error de diarización. La detección de solapamiento de locutores se presenta por tanto como una alternativa adecuada en el postprocesado de marcas.

Este enfoque permite mejorar los resultados en base a dos aspectos diferentes:

- El primer aspecto está relacionado con la selección de segmentos descrita en el punto anterior. La detección de segmentos con solapamiento de locutores permite que éstos sean excluidos en el proceso de entrenamiento de los modelos. Al evitar los segmentos con información perteneciente a varios locutores, se consigue aumentar la pureza de los clusters, obteniendo con ello modelos de locutor más robustos.

- El segundo aspecto está relacionado con la reducción del error de omisión del sistema. Los segmentos con solapamiento de locutores detectados requieren al menos dos etiquetas como salida del sistema. La detección de solapamiento permite al sistema asignar una o varias etiquetas adicionales a dichos seg-

mentos, reduciendo de esta forma el error de omisión y con ello, el error de diarización global del sistema.

En [66], la detección de solapamiento consigue una mejora relativa del 4.2 %, 4.5 % y 0.54 % en el DER respecto a tres sistemas de diarización distintos. Del mismo modo, en [159], el mismo enfoque se ha aplicado con éxito a dos sistemas de diarización diferentes, consiguiendo una mejora relativa del 6.9 % y 11.6 % del DER en escenario único (una única ubicación en todas las grabaciones) y del 10.2 % y 9.5 % del DER en escenario múltiple (dos ubicaciones distintas).

En [14], se realiza un postprocesado de los segmentos en los que se detecta solapamiento de locutores, para asignar a dichos segmentos la etiqueta de un segundo locutor, consiguiendo una mejora relativa del 7.4 % del DER en condiciones de campo cercano (micrófonos headset) y 3.6 % del DER en condiciones de campo lejano (micrófonos de sobremesa). Un enfoque similar se utiliza en [15], donde el postprocesado de los segmentos con solapamiento supone una mejora relativa del 6.8 % del error de diarización.

6.1.4 **Refinamiento de fronteras**

Por último, es habitual encontrar en los sistemas una etapa de resegmentación o refinamiento de las marcas que permita la mejora de los resultados obtenidos. Si se dispone de material de los locutores presentes en el audio es posible entrenar modelos para cada uno de ellos y utilizarlos en una nueva segmentación y refinar las fronteras establecidas por el sistema de diarización.

En [55], el refinamiento de las marcas incluye un postprocesado que modela la interacción entre los participantes, mediante el cual se consigue una mejora relativa del 14.1 % del error de diarización. Con un enfoque distinto, en [88], se lleva a cabo un postprocesado de los segmentos cortos para ajustar las fronteras de dichos segmentos o dividirlos entre los segmentos contiguos.

6.2 Técnica de mejora propuesta

Uno de los problemas que tiene que afrontar todo sistema de diarización es la correcta identificación del número de locutores presentes en la grabación. Tras la etapa de segmentación de locutores, el audio es dividido en distintos segmentos que deben ser agrupados en distintos *clusters*. En un sistema ideal, cada uno de los *clusters* generados se corresponde con uno de los locutores presentes en el audio.

Un proceso de agrupación de locutores que persigue asegurar la pureza de los *clusters* puede generar problemas de sobre-*clustering* en el sistema. Cuando esto sucede, el audio perteneciente a un locutor se reparte entre varios *clusters*, por lo que el tiempo correspondiente a los *clusters* excedentes resultará en un aumento directo del error de diarización. Del mismo modo, distintas condiciones en el fondo del audio (música o ruido) pueden dar lugar a distintos *clusters* con información perteneciente a un único locutor [162], por lo que los sistemas de diarización a menudo incluyen técnicas de normalización antes del proceso de agrupación.

El criterio de agrupación de *clusters* a menudo se muestra como un punto crítico del sistema de diarización, resultando complicado el ajuste óptimo en cada caso. El sobre-*clustering* se plantea por tanto, como un problema común en cualquier sistema de diarización. La técnica de mejora propuesta consiste en un postprocesado de las marcas proporcionadas por el sistema, que permita identificar y reagrupar los *clusters* pertenecientes a un mismo locutor, reduciendo el problema de sobre-*clustering* del sistema y por tanto, el error de diarización obtenido por el mismo.

La Figura 6.1 muestra los distintos bloques que componen la etapa de postprocesado propuesta: el bloque de refinado de la segmentación voz/no voz, el bloque de asimilación de segmentos de corta duración y por último el bloque de fusión de *clusters* correspondientes a un mismo locutor. A continuación se describe cada uno de los bloques.

6.2.1 Refinado de la segmentación voz/no voz

Tanto el Error de Omisión como el Error de Inserción se deben a un mal funcionamiento de la etapa de detección de voz. Debido a este incorrecto funcionamiento, los segmentos de música o ruido que deberían ser eliminados pasan a la etapa de agrupación de locutores, recibiendo generalmente una nueva etiqueta de locutor.

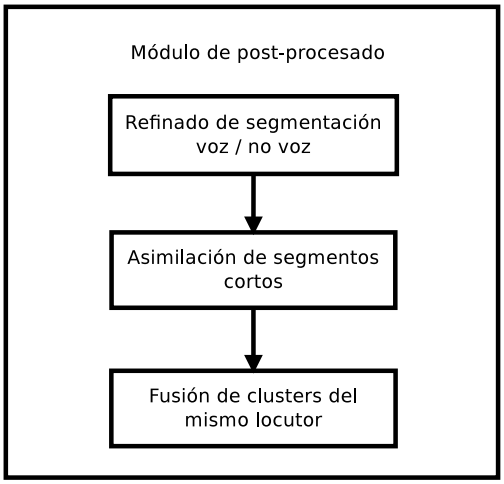


Figura 6.1: Diagrama de la etapa de postprocesado propuesta

El objetivo del primer bloque de la etapa de postprocesado es el refinado de los errores cometidos en la etapa de detección de voz. Para ello, en primer lugar, se propone realizar el entrenamiento de un modelo GMM para cada uno de los *clusters* presentes a la salida del sistema de diarización y de un modelo GMM para el silencio a partir de segmentos extraídos de las sesiones de entrenamiento. A continuación, para cada segmento de voz marcado por el sistema de diarización, se realizará una resegmentación por algoritmo de Viterbi que incluya los modelos entrenados para el silencio y para el locutor marcado originalmente en dicho segmento. Los silencios de corta duración pueden ser eliminados en este punto.

De esta forma, los silencios, música o posibles eventos acústicos detectados se marcan finalmente como “no voz”, por lo que se consigue una reducción del Error de Inserción. Por el contrario, segmentos de distintos locutores que han sido incluidos erróneamente en los diferentes *clusters* pueden ser marcados en este bloque como ”no voz”, provocando un aumento del Error de Omisión. En estos casos sin embargo, se consigue un aumento en la pureza de los *clusters* y con ello, una mejora en el rendimiento de los bloques posteriores.

6.2.2 Asimilación de segmentos cortos

El segundo bloque de la etapa de postprocesado persigue eliminar los segmentos de corta duración marcados de forma errónea cuando uno de los locutores realiza una intervención de larga duración. Para ello, en primer lugar, dichos segmentos deben ser identificados en función de su duración y la del segmento que los precede. Es necesario ajustar empíricamente los valores de las duraciones a tener en cuenta para optimizar el funcionamiento de este bloque. Se utilizarán para ello las señales de la parte de entrenamiento de la base de datos.

A continuación, de forma similar a lo dispuesto en el bloque anterior, se propone entrenar un modelo GMM a partir del audio disponible para el locutor marcado originalmente en el segmento y un modelo GMM para el locutor marcado en el segmento anterior usando el audio disponible para él en la grabación. Finalmente, si el segmento queda mejor modelado por el GMM entrenado para el locutor del segmento anterior, éste será asimilado por el *cluster* de dicho locutor adyacente.

De nuevo en estos casos se consigue un aumento en la pureza de los *clusters* y con ello, una mejora en el rendimiento del bloques de fusión de *clusters* posterior.

6.2.3 Fusión de *clusters*

El tercer y último bloque de la etapa de postprocesado es el encargado de llevar a cabo la fusión de *clusters* pertenecientes al mismo locutor. Para ello, en primer lugar se propone entrenar modelos GMM para cada uno de los *clusters* identificados por el sistema de diarización, con parte del material contenido en cada uno de ellos. A continuación, para cada uno de los diferentes *clusters*, se extraerá un segmento de audio no utilizado en el entrenamiento del modelo GMM del *cluster* y se evaluará la similitud de dicho segmento con el modelo del locutor marcado originalmente en dicho segmento y cada uno de los modelos de los *clusters* restantes. La diferencia de verosimilitudes obtenidas por los modelos de los *clusters* analizados en el segmento evaluado se propone como métrica en este caso.

Valores reducidos en la diferencia de verosimilitudes obtenidas por ambos modelos indicarán la existencia de información similar en los *clusters*. En función de dichos valores y las diferencias relativas obtenidas para cada uno de los diferentes

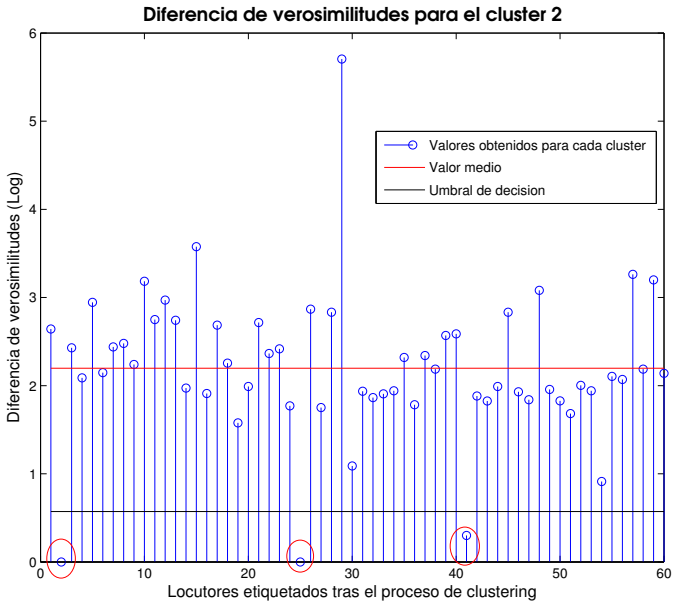


Figura 6.2: Diferencias de verosimilitudes obtenidas para un *cluster* ejemplo

clusters es posible determinar si la combinación de dos o más *clusters* resultará adecuada en cada caso. El umbral de decisión que determine si dos *clusters* deben ser fusionados deberá ser establecido empíricamente mediante la optimización de resultados de la etapa diseñada en la parte de desarrollo de la base de datos.

Supongamos un sistema de diarización que realiza la división de un audio de ejemplo en 60 *clusters* diferentes. La Figura 6.2 muestra las diferencias obtenidas tras el análisis de un segmento de audio extraído del *cluster* 2. En este caso, se puede observar cómo los *clusters* 25 y 41 obtienen valores de diferencia de verosimilitudes reducidos en relación al resto de *clusters*, por lo que es posible determinar que los *clusters* 2, 25 y 41 contienen audio perteneciente al mismo locutor. La fusión de dichos *clusters* reducirá el sobre-*clustering* generado en el sistema y en gran medida el error de diarización obtenido por el mismo.

Este bloque de fusión de *clusters* es el responsable de la reducción del over *clustering* generado en el sistema, por lo que la respuesta de la etapa de postprocesado está directamente relacionada con el buen funcionamiento del mismo. En cuanto a los dos primeros bloques, han sido diseñados para corregir errores de menor importancia y mejorar el rendimiento de este bloque.

6.3 Validación del método propuesto

Con el fin de comprobar el funcionamiento de la etapa de postprocesado diseñada se han realizado diversos experimentos [138] [139]. En primer lugar, se han utilizado las etiquetas de salida de tres sistemas de diarización diferentes junto con la base de datos proporcionada en la campaña de evaluación Albayzin 2010. Posteriormente, se ha aplicado la etapa desarrollada a uno de los sistemas de diarización base utilizando en este caso una base de datos diferente, comprobando de esta forma la capacidad de generalización de la misma.

6.3.1 Albayzin 2010

La campaña de evaluación de diarización de locutores Albayzin 2010 [158] proponía como tarea principal la diarización de noticias broadcast, utilizando para ello la base de datos del canal catalán de televisión 3/24, recogida en la sección 2.1.1. Cinco sistemas distintos desarrollados por cinco laboratorios de investigación diferentes tomaron parte en dicha campaña de evaluación.

En la figura 6.3, se describe el sistema de diarización propuesto por el grupo Aholab con motivo de su participación en la campaña de evaluación [87].

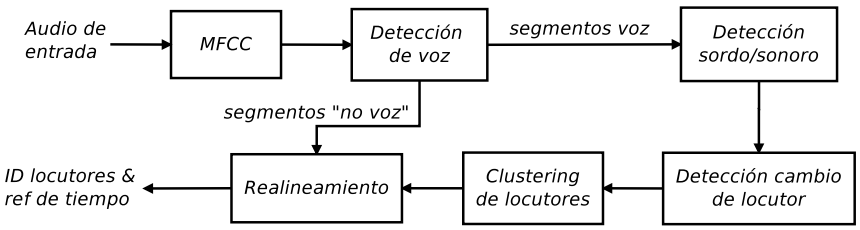


Figura 6.3: Esquema del sistema de diarización propuesto por el grupo Aholab

Dicho sistema consiste en una implementación eficiente de un detector de cambio de turno basado en BIC, que utiliza únicamente los segmentos sonoros de voz y una agrupación de locutores off-line mediante un proceso de *clustering* jerárquico acumulativo de abajo arriba. Con un error de diarización del 30 %, el sistema obtuvo el mejor resultado en la parte de test de la base de datos, proclamándose vencedor de la campaña de evaluación de locutores Albayzin 2010. A pesar de

Tabla 6.1: DER obtenido al aplicar al sistema aholab la etapa de postprocesado propuesta en las sesiones de entrenamiento de la base de datos Albayzin 2010

Sesión	DER	B1	B2	B3
1	22.17 %	21.83 %	21.54 %	29.49 %
2	24.58 %	24.46 %	24.38 %	13.10 %
3	23.10 %	23.01 %	22.92 %	18.11 %
4	27.47 %	27.67 %	27.50 %	27.50 %
5	14.15 %	12.94 %	12.93 %	9.89 %
6	21.22 %	21.40 %	21.32 %	16.21 %
7	24.84 %	24.86 %	24.89 %	27.72 %
8	27.26 %	27.38 %	27.38 %	19.90 %
9	28.92 %	28.28 %	28.60 %	26.80 %
10	34.75 %	34.54 %	35.26 %	26.80 %
11	27.94 %	27.70 %	27.90 %	15.91 %
12	27.42 %	27.22 %	27.22 %	25.54 %
13	31.92 %	32.13 %	31.86 %	32.34 %
14	41.16 %	40.87 %	41.00 %	25.84 %
15	32.50 %	32.73 %	32.62 %	27.25 %
16	32.06 %	32.09 %	32.02 %	24.18 %
1-16	28.25 %	28.14 %	28.16 %	**23.33 %**

ello, resulta evidente el margen de mejora de los resultados obtenidos por el sistema, siendo el over *clustering* uno de los principales problemas observados en el análisis de errores llevado a cabo tras la campaña de evaluación.

En primer lugar, se ha utilizado la parte de entrenamiento de la base de datos para llevar a cabo la optimización de parámetros de la etapa. En general, la optimización se ha llevado a cabo de forma homogénea, utilizando todas las señales disponibles, sin embargo, el ajuste necesario para evitar errores en la primera sesión reduce en gran medida el funcionamiento de la etapa en el conjunto de entrenamiento, por lo que dicha señal ha sido excluida del proceso de optimización.

En la Tabla 6.1 se recoge el DER obtenido tanto para las marcas originales del

Tabla 6.2: DER obtenido al aplicar al sistema aholab la etapa de postprocesado propuesta en las sesiones de test de la base de datos Albayzin 2010

Sesión	DER	B1	B2	B3
17	34.92 %	34.69 %	34.62 %	33.97 %
18	31.35 %	30.77 %	30.82 %	19.36 %
19	27.14 %	27.47 %	27.46 %	21.05 %
20	34.72 %	34.57 %	34.76 %	25.52 %
21	34.20 %	34.24 %	34.14 %	18.38 %
22	33.06 %	33.36 %	33.33 %	29.81 %
23	24.92 %	25.05 %	25.18 %	19.48 %
24	22.99 %	22.96 %	23.11 %	17.76 %
17-24	30.11 %	30.08 %	30.13 %	**23.40 %**

sistema como al aplicar cada uno de los bloques de la etapa de postprocesado a las sesiones de entrenamiento. La última fila muestra el valor de DER obtenido de forma global en la parte de entrenamiento de la base de datos.

Se puede observar cómo se ha conseguido reducir el error en prácticamente la totalidad de las sesiones. De forma global se ha obtenido una reducción del error de diarización del 17.4 % al aplicar la etapa de postprocesado al set de entrenamiento. La primera sesión, excluida del proceso de optimización, presenta no obstante un aumento significativo en el error obtenido.

En el caso de los dos primeros bloques, apenas se consigue reducción del error, sin embargo, como se ha comentado con anterioridad, permiten aumentar la pureza de los *clusters* y contribuyen a mejorar el funcionamiento del tercer bloque.

La Tabla 6.2 muestra el resultado obtenido en las sesiones de test de la base de datos. Al igual que en el caso anterior, se recogen los valores de DER tanto para las marcas originales del sistema de diarización como a la salida de cada uno de los bloques de la etapa de postprocesado, así como el valor del DER obtenido de forma global en las sesiones de test.

Se puede observar cómo en este caso se ha conseguido mejorar el resultado en la totalidad de las sesiones, con una reducción del error de diarización del 22.3 %

Tabla 6.3: DER obtenido al aplicar al sistema aholab online la etapa de postprocesado propuesta en las sesiones de entrenamiento de la base de datos Albayzin 2010

Sesión	DER	B1	B2	B3
1	28.22 %	27.79 %	21.54 %	24.74 %
2	22.30 %	22.24 %	22.08 %	15.17 %
3	24.39 %	24.24 %	24.25 %	21.67 %
4	21.32 %	21.36 %	20.77 %	20.77 %
5	17.64 %	17.19 %	16.81 %	13.52 %
6	21.00 %	21.14 %	21.14 %	14.35 %
7	23.73 %	23.78 %	23.90 %	23.07 %
8	24.91 %	25.05 %	25.16 %	25.16 %
9	27.53 %	26.91 %	27.03 %	20.21 %
10	29.27 %	29.17 %	29.15 %	13.92 %
11	25.96 %	25.83 %	25.83 %	14.88 %
12	31.33 %	31.41 %	31.49 %	29.86 %
13	25.84 %	26.00 %	26.11 %	23.29 %
14	35.82 %	35.67 %	35.73 %	21.78 %
15	27.86 %	28.11 %	28.48 %	27.04 %
16	30.58 %	30.82 %	31.02 %	26.37 %
1-16	26.77 %	26.72 %	26.76 %	**21.38 %**

en el global de las sesiones de test, lo que prueba la validez de la etapa de postprocesado propuesta. Se ha conseguido además homogeneizar el resultado del sistema en ambas partes de la base de datos, con un DER final en torno al 23 %.

La etapa de postprocesado propuesta ha conseguido reducir el error en la mayor parte de las sesiones, tanto en la parte de entrenamiento como en la de prueba. Los dos primeros bloques apenas consiguen reducir por sí solos el error obtenido originalmente por el sistema de diarización, sin embargo, contribuyen a mejorar el funcionamiento del tercer bloque, responsable de la significativa mejora obtenida al aplicar la etapa de postprocesado.

Tabla 6.4: DER obtenido al aplicar al sistema aholab online la etapa de postprocesado propuesta en las sesiones de test de la base de datos Albayzin 2010

Sesión	DER	B1	B2	B3
17	30.74 %	30.81 %	31.07 %	27.37 %
18	25.95 %	25.40 %	25.72 %	17.97 %
19	23.86 %	24.25 %	24.22 %	19.45 %
20	33.36 %	33.21 %	33.60 %	23.87 %
21	21.38 %	21.54 %	21.74 %	14.64 %
22	35.00 %	35.11 %	35.12 %	30.20 %
23	21.79 %	21.99 %	22.06 %	17.25 %
24	22.99 %	22.90 %	22.85 %	17.98 %
17-24	27.17 %	27.18 %	27.32 %	**21.45 %**

Una vez comprobado el buen funcionamiento de la etapa de postprocesado sobre el sistema de diarización propuesto por el grupo Aholab para la campaña de evaluación Albayzin 2010, se han utilizado las marcas proporcionadas por un nuevo sistema desarrollado en el laboratorio, de arquitectura similar al anterior, pero que trabaja de forma online [87]. Para realizar este nuevo experimento se han mantenido la base de datos utilizada en la campaña de evaluación Albayzin 2010 y los parámetros de configuración obtenidos para el sistema de diarización anterior usando las sesiones de entrenamiento. De esta forma evaluaremos la capacidad de generalización de la etapa de postprocesado desarrollada.

Las tablas 6.3 y 6.4 muestran los resultados obtenidos al aplicar la etapa de de postprocesado desarrollada a las marcas proporcionadas por este nuevo sistema de diarización sobre las señales de entrenamiento y evaluación de la base de datos Albayzin 2010. Al igual que en los casos anteriores, se muestran los valores de DER obtenidos para las marcas originales del sistema de diarización y a la salida de cada uno de los bloques de la etapa de postprocesado.

Se puede observar en las tablas cómo la mejora conseguida es similar a la obtenida sobre el sistema utilizado en primer lugar, con una reducción del error de diarización del 20.1 % en la parte de entrenamiento y un 21 % en la parte de test,

Tabla 6.5: DER obtenido al aplicar al sistema del grupo GTM la etapa de postprocesado propuesta en las sesiones de entrenamiento de la base de datos Albayzin 2010

Sesión	DER	B1	B2	B3
1	31.19 %	30.94 %	30.52 %	33.83 %
2	25.70 %	25.54 %	25.45 %	14.82 %
3	15.75 %	15.92 %	15.59 %	16.50 %
4	22.17 %	22.28 %	21.92 %	28.72 %
5	21.28 %	21.14 %	21.05 %	14.72 %
6	22.17 %	22.59 %	22.41 %	27.34 %
7	19.98 %	20.10 %	19.87 %	19.87 %
8	20.89 %	21.25 %	21.01 %	22.51 %
9	26.17 %	25.84 %	25.78 %	30.12 %
10	41.39 %	41.14 %	41.70 %	41.70 %
11	18.66 %	18.65 %	18.56 %	20.57 %
12	27.66 %	27.63 %	27.19 %	28.27 %
13	22.44 %	22.85 %	22.44 %	22.44 %
14	25.24 %	25.30 %	24.83 %	25.35 %
15	35.51 %	35.50 %	35.20 %	34.49 %
16	27.07 %	27.36 %	26.93 %	27.94 %
1-16	25.48 %	25.54 %	**25.31 %**	25.91 %

aunque la inserción de la etapa de postprocesado diseñada elimina en este caso la característica de funcionamiento online de este sistema. Como se ha comentado anteriormente, éste comparte en gran medida la arquitectura del sistema diseñado para la campaña Albayzin 2010, por lo que la etapa de postprocesado consigue mejorar su rendimiento a pesar de no modificar la configuración de la misma.

Por último, se ha aplicado la etapa de postprocesado a las marcas proporcionadas por un sistema de diarización desarrollado por el grupo GTM de la Universidad de Vigo [36]. En este caso, se trata de un sistema de arquitectura diferente.

La tabla 6.5 muestra los resultados obtenidos en la parte de entrenamiento de la base de datos. Una vez más, se han mantenido en este experimento los parámetros

Tabla 6.6: DER obtenido al aplicar al sistema del grupo GTM la etapa de postprocesado propuesta en las sesiones de test de la base de datos Albayzin 2010

Sesión	DER	B1	B2	B3
17	26.38 %	26.57 %	26.14 %	27.67 %
18	25.25 %	24.84 %	24.48 %	33.05 %
19	18.07 %	18.46 %	18.41 %	19.08 %
20	36.18 %	36.22 %	36.03 %	36.42 %
21	25.43 %	25.38 %	25.40 %	19.57 %
22	29.29 %	29.33 %	29.04 %	30.73 %
23	25.83 %	25.77 %	25.35 %	26.87 %
24	18.94 %	18.80 %	17.80 %	19.79 %
17-24	25.62 %	25.62 %	**25.26 %**	27.00 %

de configuración obtenidos para el sistema de diarización utilizado en primer lugar (Aholab 2010), sobre las sesiones de entrenamiento de la base de datos. Se puede observar cómo en este caso no se consigue mejora global de los resultados del sistema original al aplicar la etapa de postprocesado desarrollada. Los dos primeros bloques (refinado de segmentación voz vs. no voz y asimilación de segmentos cortos), de carácter menos específico, presentan unos resultados similares a los obtenidos en los casos anteriores. El bloque de fusión de *clusters* sin embargo, alterna sesiones de mejora significativa con sesiones de mayor error.

El objetivo de la etapa de postprocesado diseñada es corregir el posible over *clustering* en el sistema de diarización, por lo que el resultado obtenido al aplicar esta etapa a sistemas que no presentan este problema puede ser el contrario al que se busca. No obstante, cabe recordar que no se han modificado los parámetros de configuración de la etapa de postprocesado, por lo que el ajuste de los mismos a este nuevo sistema de diarización podría evitar fusiones erróneas de *clusters* que aumentan significativamente el error obtenido en gran parte de las señales. El buen resultado obtenido, con una amplia reducción del error de diarización en algunas de las señales, como las sesiones 2 y 5, apuntan también en la misma dirección.

La tabla 6.6 muestra los resultados obtenidos en la parte de evaluación de la

base de datos. Se puede observar cómo las conclusiones obtenidas son similares a las extraídas en la parte de entrenamiento. No se ha conseguido una mejora de los resultados originales, sin embargo, la optimización de parámetros podría mejorar el rendimiento de la etapa de postprocesado.

6.3.2 Experimentos sobre otra base de datos

En este caso se ha propuesto estudiar la independencia de la etapa de postprocesado desarrollada de la base de datos utilizada, por lo que se ha utilizado de nuevo el sistema de diarización propuesto por el grupo Aholab para la campaña de evaluación Albayzin 2010 para marcar dos señales pertenecientes a la base de datos Ahonews, recogida en la sección 2.2.1. Más concretamente, se han utilizado las dos primeras sesiones artificiales creadas a partir de archivos correspondientes a emisiones de noticias de la Radiotelevisión Vasca (EiTB).

La primera de las señales utilizada es una sesión de 20 minutos de duración, en la que aparecen 9 locutores diferentes que realizan largas intervenciones en condiciones de bajo ruido. Las características de esta señal favorecen a priori el rendimiento de los dos primeros bloques de la etapa de postprocesado.

La segunda señal es una sesión de 25 minutos de duración, en la que 40 locutores alternan cortas intervenciones, y que incluye segmentos con ruido y música de fondo, por lo que el funcionamiento del tercer bloque debería tener mayor relevancia en este caso. Ninguna de las señales incluye habla solapada.

El resultado obtenido se recoge en la tabla 6.7. Podemos observar cómo la mejora de los resultados obtenida es similar a la conseguida sobre la base de datos Albayzin 2010. En este caso se ha obtenido una reducción del error de diarización del 17 %. Estos resultados demuestran que la etapa de postprocesado desarrollada es aplicable a bases de datos de diferentes características.

A pesar de haber comprobado el buen funcionamiento de la etapa diseñada para la base de datos utilizada, el resultado obtenido para el tercer bloque de la etapa sí muestra dependencia con el tipo de señal introducido en el sistema. Señales con elevado nivel de ruido o música de fondo (como sucede en la segunda sesión), favorecen el over *clustering* generado por el sistema de diarización original, por lo que el rendimiento de la etapa de postprocesado, diseñada específicamente para

Tabla 6.7: DER obtenido al aplicar el postprocesado al sistema diarización propuesto por el grupo Aholab en la base de datos de EiTB

Sesión	DER	B1	B2	B3
1	35.65 %	34.65 %	32.30 %	32.30 %
2	26.83 %	26.78 %	26.78 %	20.53 %
1-2	30.26 %	29.84 %	28.93 %	**25.11 %**

reducir este problema, resulta notable en este caso. Por el contrario, con señales con bajo nivel de ruido sólo los dos primeros bloques resultan efectivos, por lo que el rendimiento de la etapa de postprocesado se reduce considerablemente.

En principio, el bloque de fusión de *clusters* no debería introducir nuevos errores, pero para obtener cierta mejora en los resultados obtenidos por el sistema de diarización original, es necesario realizar la optimización de parámetros de configuración de la etapa de postprocesado en función de la base de datos a utilizar (número de locutores presente en las señales, música, ruido de fondo...).

6.4 Conclusiones

Se han analizado en este capítulo las distintas técnicas de mejora de la diarización basadas en el postprocesado y análisis de la información proporcionada por un sistema de diarización base.

En primer lugar, se han identificado en la literatura distintos enfoques destinados a la mejora de la diarización basados en el análisis de las marcas proporcionadas por un sistema de diarización base, mostrando especial interés en aquellos orientados a la mejora de la segmentación voz, la selección de segmentos, la detección de voz solapada y el refinamiento de fronteras.

A continuación, se ha descrito la técnica propuesta para la mejora de la diarización de locutores, cuyo objetivo principal consiste en reducir el problema de over *clustering* presente en algunos sistemas, y con ello, el error de diarización obtenido por los mismos. Se ha diseñado para ello una etapa de procesado compuesta por tres bloques destinados a combatir distintos tipos de error cometidos por el sistema: refinado de la segmentación voz/no voz, asimilación de los segmentos cortos y fusión de los *clusters* pertenecientes al mismo locutor.

Por último, se han realizado diversos experimentos con el fin de comprobar el funcionamiento de la etapa de postprocesado diseñada, utilizando para ello las etiquetas de salida de tres sistemas de diarización diferentes junto con la base de datos Albayzin 2010. Adicionalmente, se ha aplicado la etapa desarrollada a uno de los sistemas de diarización utilizando en este caso la base de datos Ahonews, para comprobar la capacidad de generalización de la misma.

En primer lugar se han mostrado los resultados obtenidos mediante la aplicación de la etapa propuesta al sistema de diarización desarrollado por el grupo Aholab con motivo de su participación en la campaña de evaluación Albayzin 2010. En este caso se ha obtenido una reducción del error de diarización del 17.4 % en las sesiones de entrenamiento y del 22.3 % en las sesiones de evaluación, lo que demuestra la validez de la etapa de postprocesado propuesta.

Del mismo modo, se han presentado los resultados obtenidos al aplicar la etapa propuesta a un nuevo sistema desarrollado en el laboratorio Aholab, de arquitectura similar al anterior, manteniendo la base de datos utilizada en la campaña de evaluación Albayzin 2010. De forma similar a lo ocurrido con el sistema de diarización

utilizado en primer lugar, se ha obtenido una reducción del error de diarización del 20.1 % en la parte de entrenamiento y del 21 % en la parte de evaluación, demostrando nuevamente el buen funcionamiento de la etapa de postprocesado propuesta.

No se ha obtenido mejora sin embargo, al aplicar la etapa de postprocesado al sistema de diarización desarrollado por el grupo GTM de la Universidad de Vigo, de arquitectura muy diferente a los anteriores. La etapa propuesta alterna en este caso sesiones de mejora significativa con sesiones de mayor error, por lo que se plantea la posibilidad de mejorar los resultados del sistema mediante la optimización de los parámetros de configuración de los bloques que forman la etapa.

Por último, se ha demostrado la capacidad de generalización de la etapa de postprocesado mediante la aplicación de la misma al sistema de diarización Aholab-Albayzin 2010, utilizando en este caso distintas señales de la base de datos datos Ahonews. En este caso se ha obtenido una reducción general del error de diarización del 17 %, resultando visible la dependencia del correcto funcionamiento de la etapa de determinadas características de las señales (número de locutores, música de fondo, ruido...), lo que demuestra la necesidad de optimización de los parámetros de configuración para la obtención de un mejor resultado.

La felicidad es saber unir el final con el principio.

Pitágoras

CAPÍTULO

7

Conclusiones

Esta tesis presenta distintos aspectos involucrados en el desarrollo de un sistema de diarización de locutores, centrando el esfuerzo en los problemas más habituales en los diferentes campos de aplicación de este tipo de sistemas.

Durante el desarrollo de este trabajo se han diseñado distintos algoritmos y técnicas de mejora de la diarización de locutores, con resultados favorables tanto en entorno de audio *broadcast* como en entorno de reuniones de trabajo. Otro aspecto importante en el desarrollo de esta tesis ha sido la participación en distintas campañas de evaluación, que han permitido realizar una evaluación contrastable de los distintos sistemas implementados.

En este capítulo se recogen las principales conclusiones extraídas durante el desarrollo de esta tesis. En primer lugar, se resumen las aportaciones realizadas en las distintas áreas de la diarización examinadas, así como posibles trabajos futuros derivados de este estudio. Finalmente, se presenta la difusión de resultados llevada a cabo durante el desarrollo de este trabajo relativa a publicaciones y conferencias.

7.1 Aportaciones de la tesis y trabajos futuros

Bases de datos

Se ha llevado a cabo la grabación y recopilación de dos bases de datos en castellano y euskera, diseñadas para el desarrollo de tareas de diarización de locutor en dos entornos de aplicación diferentes: difusión de audio y reuniones de trabajo.

La mayor parte de los estudios realizados en los últimos años dedicados a recopilar y transcribir diferentes bases de datos, ya sea en el entorno de reuniones de trabajo o de difusión de audio, centran sus esfuerzos en la obtención de grabaciones monolingües. Las bases de datos que incluyen habla en más de un idioma a menudo presentan diferentes locutores para los distintos idiomas o han sido diseñadas para realizar tareas en otras áreas del procesado de voz, como el reconocimiento del habla o el reconocimiento de locutor.

Con la creación de estas nuevas bases de datos de voz se ha tratado de llenar este vacío, permitiendo el desarrollo y la implementación de sistemas de diarización multilingües en los dos ámbitos de aplicación mencionados. El trabajo realizado en este sentido ha quedado recogido en [142].

Segmentación de audio

La segmentación de audio a menudo forma parte de sistemas más complejos en otras áreas del procesado de la voz, como el reconocimiento automático del habla o el reconocimiento de locutores, por lo que un mejor tratamiento previo del audio resulta necesario para mejorar el rendimiento de este tipo de sistemas.

Uno de los problemas de la segmentación de audio broadcast reside en la correcta identificación de los segmentos que contienen música de fondo de bajo nivel, que a menudo pasa desapercibida y no es detectada por los sistemas de segmentación. Por ello, se ha diseñado una técnica de segmentación basada en el postprocesado de segmentos de voz en busca de música de fondo de bajo nivel. Se trata de un método sencillo pero adecuado a las necesidades de este tipo de audio broadcast.

Se ha diseñado e implementado un sistema de segmentación de audio broadcast basado en la técnica propuesta. Mediante el postprocesado de los segmentos de voz,

este sistema es capaz de identificar aquellos con música de fondo de bajo nivel, reduciendo considerablemente el Error de Omisión para la clase de música.

Se ha comprobado el buen funcionamiento del sistema mediante la participación en la campaña de evaluación de segmentación de audio Albayzin 2012, donde el sistema desarrollado ha obtenido el mejor resultado de la evaluación en la que tomaron parte 6 sistemas desarrollados por 5 grupos de investigación diferentes. En este caso, el postprocesado de los segmentos ha propiciado una reducción relativa del 27 % en el error de la clase de música.

Por otra parte, entornos más complejos requieren una mejor clasificación de los segmentos de voz a la hora de llevar a cabo una segmentación más precisa. Se ha diseñado por tanto una nueva técnica de segmentación basada en clasificación de i-vectors que permite el refinamiento de los segmentos de voz para determinar la presencia de música y/o ruido de fondo en el audio.

Igualmente, se ha diseñado e implementado un segundo sistema de segmentación de audio broadcast basado en la nueva técnica propuesta que ha mostrado un rendimiento claramente superior al de sistemas tradicionales basados en HMMs.

Por último, se ha comprobado el buen funcionamiento del sistema mediante la participación en la campaña de evaluación de segmentación de audio Albayzin 2014, donde el nuevo sistema desarrollado ha obtenido el mejor resultado de la evaluación en la que tomaron parte 7 sistemas desarrollados por 4 grupos de investigación diferentes. El sistema basado en i-vectors ha demostrado su buen rendimiento frente al sistema basado en HMMs con un 22.47 % y un 27.37 % de SER respectivamente, mostrando la validez de la técnica de segmentación propuesta.

En el futuro se propone la realización del análisis de otro tipo de parámetros no considerados en este estudio, que permitan una mejor discriminación entre música y ruido, principales fuentes de error de los sistemas de segmentación. Además de contrastar su comportamiento frente al resto de parametrizaciones utilizadas, se plantea la posibilidad de desarrollar sistemas basados en las distintas parametrizaciones y utilizar sus resultados para obtener un mejor rendimiento global mediante técnicas de fusión de clasificadores.

Diversas publicaciones han recogido los resultados de la investigación llevada a cabo en este ámbito, a través tanto de la participación en congresos [137] [141], como por medio de artículos en revistas especializadas [22].

Fusión de clasificadores

La fusión de etiquetas resulta interesante en múltiples problemas del mundo real: unificación de la información de anotadores expertos, sistemas colaborativos entre laboratorios, fusión de clasificadores de muy diversa naturaleza... Los distintos métodos de fusión de etiquetas, sin embargo, tienden a realizar clasificaciones muy pobres cuando las bases de datos presentan desequilibrio entre sus clases, algo que ocurre habitualmente en el área del procesado de la voz.

Por ello, se ha realizado un análisis de distintos métodos de fusión de etiquetas, identificando en cada caso los problemas que presentan, prestando especial interés al comportamiento de cada uno en condiciones de desequilibrio en la base de datos.

Se ha presentado una nueva técnica de fusión de etiquetas que tiene en cuenta el mencionado desequilibrio entre clases de la base de datos. Este nuevo método, además del número de casos almacenado durante la etapa de entrenamiento, tiene en cuenta lo representativos que resultan esos casos en la base de datos.

Se han realizado además múltiples experimentos con el objetivo de validar el funcionamiento del algoritmo propuesto frente a otros métodos del estado del arte. Los resultados obtenidos han mostrado un mejor comportamiento de la técnica propuesta en condiciones de desequilibrio en la base de datos, debido a que realiza una mejor clasificación de las clases minoritarias en la base de datos.

También se ha diseñado una alternativa al método propuesto con el fin de extender su funcionamiento a un nivel de score, más adecuado en el área de del reconocimiento y verificación de locutores. Se han presentado diversos experimentos realizados en este ámbito con buenos resultados de la nueva técnica propuesta.

Como trabajo futuro se plantea la posibilidad de generalizar este enfoque para abordar tareas de verificación de locutor con listas abiertas (open set). A pesar del buen funcionamiento demostrado en los experimentos realizados en este ámbito, la información utilizada por el método propuesto, concretamente la identidad de los locutores impostores en cada una de las pruebas, supone la infracción de las normas que rigen las evaluaciones en este campo, por lo que se precisa de un nuevo enfoque en la etapa de entrenamiento del método, que permita extrapolar el conocimiento de los locutores de manera individual.

Resultados preliminares de los diversos trabajos llevados a cabo en este ámbito se han recogido en [140], si bien la publicación del análisis detallado del método de fusión propuesto continúa siendo un objetivo destacado.

Segmentación de locutores

Un sistema tradicional de segmentación de locutores habitualmente requiere la disponibilidad de la base de datos completa al empezar el procesado, por lo que su utilización online, con reducidas latencias permitidas, resulta imposible.

Se ha propuesto una nueva técnica de segmentación basada en un análisis trama a trama, donde el retardo introducido sea equivalente al tamaño de ventana utilizado durante el procesado del audio. La mayoría de bases de datos de diarización en entorno de reuniones cuentan con grabaciones multicanal, por lo que el método propuesto hace uso de la correlación entre las señales de los distintos micrófonos, identificando posibles cambios de turno con un bajo retardo de procesamiento.

También se han realizado distintos experimentos para validar la técnica de segmentación propuesta mediante dos bases de datos ampliamente utilizadas en el entorno de reuniones de trabajo, contrastando el funcionamiento del método propuesto frente a una técnica clásica basada en distancia BIC, una de las más ampliamente utilizadas en la literatura en este ámbito.

Los experimentos realizados en ambos casos han mostrado mejores resultados por parte de la técnica de segmentación propuesta en lo referente a las distintas métricas de evaluación utilizadas, detectando correctamente mayor número de cambios locutor en la mayor parte de las señales utilizadas.

Como trabajo futuro, se propone diseñar un proceso de *clustering* online que complemente el método de segmentación propuesto, mediante la aplicación de distintas técnicas que permitan identificar los distintos locutores presentes en una reunión de trabajo real. Por otro lado, el número de errores de inserción cometidos por la técnica propuesta, requiere, además de un correcto funcionamiento del VAD, que el proceso de *clustering* diseñado sea capaz de prestar especial atención a los locutores identificados con anterioridad, reduciendo en la medida de lo posible el número de falsos cambios de turno detectados. La publicación del método de segmentación propuesto continúa siendo un objetivo futuro en este caso.

Postprocesado de marcas

A menudo resulta interesante la introducción de etapas de resegmentación o postprocesado, con un bajo impacto computacional, destinadas a solucionar problemas concretos de los sistema de diarización.

Uno de los problemas presentes en los sistemas de diarización es el sobre-*clustering*, que aparece como resultado de un proceso inadecuado de agrupación de *clusters*, cuyo ajuste óptimo puede resultar complicado en determinados casos.

Por ello, se ha propuesto una técnica de mejora de la diarización basada en el postprocesado de las marcas proporcionadas por el sistema, que permite identificar y reagrupar los *clusters* pertenecientes a un mismo locutor.

Se ha diseñado e implementado una etapa de postprocesado compuesta por tres bloques distintos: el bloque de refinado de la segmentación voz/no voz, el bloque de asimilación de segmentos de corta duración y por último el bloque de fusión de *clusters* correspondientes a un mismo locutor.

Se han realizado además diversos experimentos con el fin de comprobar el funcionamiento de la etapa de postprocesado diseñada, utilizando para ello las etiquetas de salida de tres sistemas de diarización y dos bases de datos diferentes.

A partir de los resultados obtenidos se ha demostrado el buen funcionamiento de la etapa de postprocesado desarrollada, con reducciones del error de diarización en torno al 20 % en los casos más favorables.

Se ha demostrado la capacidad de generalización de la etapa de postprocesado mediante la utilización de distintas bases de datos. Sin embargo, resulta visible la dependencia del buen funcionamiento de la etapa de determinadas características de las señales utilizadas (número de locutores, música de fondo, ruido...), por lo que en un futuro se plantea la implementación de un bloque previo de análisis del audio contenido en la base de datos, que permita la optimización automática de los distintos parámetros de configuración de los bloques restantes.

De nuevo, el trabajo realizado en este ámbito ha quedado recogido en diversas publicaciones [138] [139].

7.2 Difusión de resultados

A continuación se enumeran las contribuciones a revistas y conferencias internacionales, presentadas durante el desarrollo de este trabajo.

ARTÍCULOS DE REVISTA

2015 Diego Castán, David Tavarez, Paula Lopez-Otero, Javier Franco-Pedroso, Héctor Delgado, Eva Navas, Laura Docio-Fernández, Daniel Ramos, Javier Serrano, Alfonso Ortega, Eduardo Lleida, *"Albayzín-2014 evaluation: audio segmentation and classification in broadcast news domains"*, EURASIP Journal on Audio, Speech, and Music Proc., vol. 2015, no. 1, pp. 1-9, 2015.

2013 David Tavarez, Eva Navas, Daniel Erro, Ibon Saratxaga, Inma Hernáez, *"Nueva técnica de fusión de clasificadores aplicada a la mejora de la segmentación de audio."*, Procesamiento Lenguaje Natural, vol. 51, pp. 161-168, 2013.

2012 David Tavarez, Eva Navas, Daniel Erro, Ibon Saratxaga, Inma Hernáez, *"Técnicas de post-procesado de resultados en un sistema de diarización de locutores"*, Procesamiento Lenguaje Natural, vol. 49, pp. 109-116, 2012.

PARTICIPACIÓN EN CONGRESOS

2014 David Tavarez, Eva Navas, Daniel Erro, Ibon Saratxaga, Inma Hernaez, *"New Bilingual Speech Databases for Audio Diarization"*, In Proceedings of the Ninth International Conference on Language Resources and Evaluation, European Language Resources Association (ELRA), Reykjavik, Iceland, 2014.

2014 D. Tavarez, E. Navas, A. Alonso, D. Erro, I. Saratxaga, I. Hernaez, *"Aholab Audio segmentation system for albayzin 2014 evaluation campaign"*, In IberSPEECH 2014, Las Palmas de Gran Canaria, Spain, pp. 273-282, 2014.

2012 David Tavarez, Eva Navas, Daniel Erro, Ibon Saratxaga, *"Strategies to Improve a Speaker Diarisation Tool"*, In Proceedings of the Eight International Conference on Language Resources and Evaluation, European Language Resources Association (ELRA), Istanbul, Turkey, 2012.

2012 David Tavarez, Eva Navas, Daniel Erro, Ibon Saratxaga, *"Audio Segmentation System by Aholab for Albayzin 2012 Evaluation Campaign"*, In IberSPEECH 2012, Madrid, Spain, pp. 577-584, 2012.

7.3 Participación en campañas de evaluación

ALBAYZIN (DIARIZATION, AUDIO SEGMENTATION):

2014 *Best system in the Albayzin audio segmentation evaluation campaign*, In IberSPEECH 2014, Las Palmas de Gran Canaria, Spain.

2012 *Best system in the Albayzin audio segmentation evaluation campaign*, In IberSPEECH 2012, Madrid, Spain.

Bibliografía

[1] Ajmera, J., & Wooters, C. 2003 (0). A Robust Speaker Clustering Algorithm. *In: IEEE Automatic Speech Recognition Understanding Workshop*. IDIAP-RR 03-38. 131

[2] Albus, J. E., Anderson, R. H., Brayer, J. M., DeMori, R., Feng, H., Horowitz, S. L., Moayer, B., Pavlidis, T., Stallings, W. W., Swain, P. H., *et al.* . 2012. *Syntactic pattern recognition, applications*. Vol. 14. Springer Science & Business Media. 66

[3] Altınçay, H. 2005. On naive Bayesian fusion of dependent classifiers. *Pattern Recognition Letters*, **26**(15), 2463–2473. 67, 71, 99

[4] Angera, X. 2006 (Octubre). *Robust Speaker Diarization for meetings*. Ph.D. thesis, Departament de Teoria del Senyal i Comunicacions, Universitat Politécnica de Catalunya. 4

[5] Anguera, X., Woofers, C., & Hernando, J. 2005 (Nov). Speaker diarization for multi-party meetings using acoustic fusion. *Pages 426–431 of: IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.* 134

[6] Anguera, X., Wooters, C., & Hernando, J. 2007. Acoustic Beamforming for Speaker Diarization of Meetings. *IEEE Transactions on Audio, Speech, and Language Processing*, **15**(7), 2011–2022. 133

[7] Atal, B. S. 1972. Automatic Speaker Recognition Based on Pitch Contours. *The Journal of the Acoustical Society of America*, **52**(6B), 1687–1697. 28

[8] Barras, C., Gauvain, J.-L., Meignier, S., & Zhu, X. 2004 (7-10 Nov). Improving Speaker Diarization. *In: RTÓ4F Workshop*. 6, 156, 157

[9] Barras, C., Zhu, X., Meignier, S., & Gauvain, J.-L. 2006. Multi-stage speaker diarization of broadcast news. *IEEE Transactions on Audio, Speech and Language Processing*. 157

[10] Batliner, A., Zeißler, V., Frank, C., Adelhardt, J., Shi, R. P., & Elmar, N. 2003. We are not amused - but how do you know? User states in a multi-modal dialogue system. *Pages 733–736 of: Interspeech 2003*. 78

[11] Batliner, A., Hacker, C., Steidl, S., Nöth, E., DÁrcy, S., Russel, M., & Wong, M. 2004. 'You Stupid Tin Box'-Children Interacting with the AIBO Robot: A Cross-linguistic Emotional Speech Corpus. *Pages 171–174 of: Proceedings of LREC*. 95

[12] Baum, D., Schneider, D., Bardeli, R., Schwenninger, J., Samlowski, B., Winkler, T., & Köhler, J. 2010. DiSCo-a German evaluation corpus for challenging problems in the broadcast domain. *Pages 1695–1699 of: 7th International conference on Language Resources and Evaluation (LREC)*. 18

[13] Ben-Harush, O., Guterman, H., & Lapidot, I. 2008 (Dec). Pure segment selection as speaker diarization post-processing. *Pages 461–465 of: Electrical and Electronics Engineers in Israel, 2008. IEEEI 2008. IEEE 25th Convention of*. 6, 156, 158

[14] Boakye, K., Trueba-Hornero, B., Vinyals, O., & Friedland, G. 2008a. Overlapped speech detection for improved speaker diarization in multiparty meetings. *Pages 4353–4356 of: Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE. 159

[15] Boakye, K., Vinyals, O., & Friedland, G. 2008b. Two's a crowd: improving speaker diarization by automatically identifying and excluding overlapped speech. *Pages 32–35 of: INTERSPEECH 2008, 9th Annual Conference of the International Speech Communication Association, Brisbane, Australia, September 22-26, 2008*. 159

[16] Breiman, L. 1996. Bagging predictors. *Machine Learning*, **24**(2), 123–140. 68

[17] Brummer, N., Burget, L., Cernocky, J. H., Glembek, O., Grezl, F., Karafiat, M., van Leeuwen, D. A., Matejka, P., Schwarz, P., & Strasheim, A. 2007. Fusion of Heterogeneous Speaker Recognition Systems in the STBU Submission for the NIST Speaker Recognition Evaluation 2006. *Trans. Audio, Speech and Lang. Proc.*, **15**(7), 2072–2084. 117

[18] Burger, S., MacLaren, V., & Yu, H. 2002. The ISL meeting corpus: the impact of meeting type on speech style. *Pages 301–304 of: 7th International Conference on Spoken Language Processing (ICSLP)*. 18

[19] Campbell, W. M. 2008 (March). A covariance kernel for svm language recognition. *Pages 4141–4144 of: 2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. 39

[20] Campbell, W. M., Sturim, D. E., & Reynolds, D. A. 2006. Support vector machines using GMM supervectors for speaker verification. *IEEE Signal Processing Letters*, **13**(5), 308–311. 36, 39

[21] Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., Lathoud, G., Lincoln, M., Lisowska, A., McCowan, I., Post, W., Reidsma, D., & Wellner, P. 2006. The AMI Meeting Corpus: A Pre-announcement. *Pages 28–39 of: Machine Learning for Multimodal Interaction Lecture Notes in Computer Science Volume 3869*. Edinburgh, UK: Springer-Verlag. 17, 18, 140

[22] Castán, D., Tavarez, D., Lopez-Otero, P., Franco-Pedroso, J., Delgado, H., Navas, E., Docio-Fernández, L., Ramos, D., Serrano, J., Ortega, A., & Lleida, E. 2015. Albayzín-2014 evaluation: audio segmentation and classification in broadcast news domains. *EURASIP Journal on Audio, Speech, and Music Processing*, **2015**(1), 1–9. 14, 46, 66, 177

[23] Cettolo, M., & Vescovi, M. 2003 (April). Efficient audio segmentation algorithms based on the BIC. *Pages 537–540 of: International Conference on Acoustics, Speech, and Signal Processing (ICCASP 03)*, vol. 6. 131

[24] Chawla, N., Bowyer, K., Hall, L., & Kegelmeyer, W. 2002. SMOTE : Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, **16**, 321–357. 68

[25] Chen, L., Mao, X., Xue, Y., & Cheng, L. L. 2012. Speech emotion recognition: Features and classification models. *Digital Signal Processing*, **22**(6), 1154–1160. 95

[26] Chen, S. S., & Gopalakrishnan, P. S. 1998. Speaker, environment and channel change detection and clustering via the Bayesian information criterion. *Pages 127–132 of: DARPA speech recognition workshop*, vol. 6. 5, 24, 131

[27] Cieri, C., Campbell, J., Nakasone, H., Miller, D., & Walker, K. 2004. The Mixer Corpus of Multilingual, Multichannel Speaker Recognition Data. *Pages 26–28 of: Proc. 4th International Conference on Language Resources and Evaluation (LREC)*. 18

[28] Cortes, C., & Vapnik, V. 1995. Support-vector networks. *Machine Learning*, **20**(3), 273–297. 37

[29] Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., & Taylor, J. G. 2001. Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine*, **18**(1), 32–80. 95

[30] Dehak, N. 2009. *Discriminative and Generative Approaches for Long- and Short-term Speaker Characteristics Modeling: Application to Speaker Verification*. Ph.D. thesis. AAINR50490. 43

[31] Dehak, N. 2011. Language recognition via i-vectors and dimensionality reduction. *In: in Interspeech*. 44

[32] Dehak, N., Kenny, P., Dehak, R., Glembek, O., Dumouchel, P., Burget, L., Hubeika, V., & Castaldo, F. 2009 (April). Support vector machines and Joint Factor Analysis for speaker verification. *Pages 4237–4240 of: 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. 42

[33] Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., & Ouellet, P. 2011. Front-End Factor Analysis for Speaker Verification. *Trans. Audio, Speech and Lang. Proc.*, **19**(4), 788–798. 42, 44

[34] Delacourt, P., & Wellekens, C.J. 2000. DISTBIC: A speaker-based segmentation for audio data indexing. *Speech Communication*, **32**(1–2), 111 – 126. 6

[35] D'Haro, L. F., Cordoba, R., Salamea, C., & Echeverry, J. D. 2014 (May). Extended phone log-likelihood ratio features and acoustic-based i-vectors for language recognition. *Pages 5342–5346 of: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 44

[36] Docio, L., Lopez, P., & Garcia, C. 2010 (November). The UVigo-GTM Speaker Diarization System for the Albayzin'10 Evaluation. *Pages 401–404 of: VI Jornadas en Tecnología del Habla and II Iberian SLTech Workshop, (FALA 2010)*. 169

[37] Duda, R.O., Hart, P.E., & Stork, D.G. 2012. *Pattern Classification*. Wiley. 39

[38] El Hannani, A., & Hennebert, J. 2008. A Review of the Benefits and Issues of Speaker Verification Evaluation Campaigns. *Pages 29–34 of: Proceedings of the ELRA Workshop on Evaluation at LREC 08, Marrakech, Morocco*. http://www.lrec-conf.org/proceedings/lrec2008/. 11

[39] Elkan, C. 2001. The foundations of cost-sensitive learning. *IJCAI International Joint Conference on Artificial Intelligence*, 973–978. 68

[40] Ellis, D. P. W., & Liu, J. C. 2004. Speaker Turn Segmentation Based on Between-Channel Differences. *In:* Garofolo, John (ed), *Proceedings of the NIST Rich Transcription Workshop*. 133

[41] Elomaa, T., & Kääriäinen, M. 2001. An analysis of reduced error pruning. *Journal of Artificial Intelligence Research*, **15**, 163–187. 88

[42] Farrell, K. R., Ramachandran, R. P., & Mammone, R. J. 1998 (May). An analysis of data fusion methods for speaker verification. *Pages 1129–1132 vol.2 of: Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 2. 117

[43] Faundez-Zanuy, M., & Monte-Moreno, E. 2005. State of the art in speaker recognition. *Aerospace and Electronic Systems Magazine, IEEE*, **20**(5), 7 –12. 2, 4

[44] Fernández, A., García, S., del Jesus, M. J., & Herrera, F. 2008. A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. *Fuzzy Sets and Systems*, **159**, 2378–2398. 87

[45] Fierrez-Aguilar, J., Garcia-Romero, D., Ortega-Garcia, J., & Gonzalez-Rodriguez, J. 2005. *Speaker Verification Using Adapted User-Dependent Multilevel Fusion*. Berlin, Heidelberg: Springer Berlin Heidelberg. Pages 356–365. 117

[46] Font, F., Roma, G., & Serra, X. 2013. Freesound Technical Demo. *Pages 411–412 of: Proceedings of the 21st ACM International Conference on Multimedia*. MM '13. New York, NY, USA: ACM. 14, 110

[47] Freund, Y., & Schapire, R. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, **55**(1), 119 – 139. 68

[48] Furui, S. 2000. *Digital Speech Processing: Synthesis, and Recognition, Second Edition,*. Signal Processing and Communications. Taylor & Francis. 27

[49] Galar, M., Fernández, A., Barrenechea, E., Bustince, H., & Herrera, F. 2012. A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Trans.*, **42**(4), 463–484. 67, 77

[50] Greenberg, C., Bansé, D., Doddington, G., Garcia-Romero, D., Godfrey, J., Kinnunen, T., Martin, A., McCree, A., Przybocki, M., & Reynolds, D. 2014.

The NIST 2014 speaker recognition i-vector machine learning challenge. *In: Odyssey: The Speaker and Language Recognition Workshop*. 16

[51] Gu, Q., Zhu, L., & Cai, Z. 2009. Evaluation measures of the classification performance of imbalanced data sets. *Pages 461–471 of: 4th International symposium, ISICA 2009, Communications in Computer and Information Science, vol. 51*. 75

[52] Gupta, V., Kenny, P., Ouellet, P., & Stafylakis, T. 2014 (May). I-vector-based speaker adaptation of deep neural networks for French broadcast audio transcription. *Pages 6334–6338 of: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 55

[53] H. Shen, Wu, C., Hsu, Y., & Chun-Shan, Y. 2011. CECOS: A Chinese-English code-switching speech database. *Pages 120–123 of: International Conference on Speech Database and Assessments (Oriental COCOSDA)*. 18

[54] Hall, M., National, H., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, **11**(1), 10–18. 59, 86

[55] Han, K., & Narayanan, S. 2009. Improved speaker diarization of meeting speech with recurrent selection of representative speech segments and participant interaction pattern modeling. *Pages 1067–1070 of: INTERSPEECH 2009, 10th Annual Conference of the International Speech Communication Association, Brighton, United Kingdom, September 6-10, 2009*. 156, 159

[56] Hand, D., & Till, R. 2001. A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. *Machine Learning*, **45**(2), 171–186. 78

[57] He, H., & Garcia, E. 2009. Learning from imbalanced data. *IEEE Trans. Knowledge and Data Eng.*, **21**(9), 1263–1284. 67

[58] Hinton, G., Deng, L., Yu, D., Dahl, G. E., r. Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., & Kingsbury, B. 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared

Views of Four Research Groups. *IEEE Signal Processing Magazine*, **29**(6), 82–97. 41

[59] Ho, T. 1998. The Random Subspace Method for Constructing Decision Forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, **20**(8), 832–844. 68

[60] Hornik, K., Stinchcombe, M., & White, H. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks*, **2**(5), 359 – 366. 24, 39

[61] Hu, G. *100 non-speech environmental sounds*. Available online: `http://web.cse.ohio-state.edu/pnl/corpus/HuNonspeech/HuCorpus.html`. 14, 110

[62] Hu, H., Xu, M. X., & Wu, W. 2007 (April). GMM Supervector Based SVM with Spectral Features for Speech Emotion Recognition. *Pages IV–413–IV–416 of: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 4. 36

[63] Hu, M., Sharma, D., Doclo, S., Brookes, M., & Naylor, P. A. 2015 (April). Speaker change detection and speaker diarization using spatial information. *Pages 5743–5747 of: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 134

[64] Huang, Y.S., & Suen, C.Y. 1993. The Behavior-Knowledge Space method for combination of multiple classifiers. *Pages 347 – 352 of: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 67, 73

[65] Huang, Y.S., & Suen, C.Y. 1995. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **17**(1), 90–94. 74

[66] Huijbregts, M., van Leeuwen, D., & de Jong, F. 2009. Speech overlap detection in a two-pass speaker diarization system. *Pages 1063–1066 of: INTERSPEECH 2009, 10th Annual Conference of the International Speech Communication Association, Brighton, United Kingdom, September 6-10, 2009*. 156, 159

[67] Imseng, David, Bourlard, Hervé, Caesar, Holger, Garner, Philip N., Lecorvé, Gwénolé, & Nanchen, Alexandre. 2012. MediaParl: Bilingual mixed language accented speech database. *Pages 263–268 of: IEEE Spoken Language Technology Workshop (SLT)*. 18

[68] Jain, A. K., & Ross, A. 2004. Multibiometric Systems. *Commun. ACM*, **47**(1), 34–40. 65

[69] Jain, A. K., Mao, J., & Mohiuddin, K. M. 1996. Artificial neural networks: a tutorial. *Computer*, **29**(3), 31–44. 40

[70] Janin, A., Baron, A., Edwards, J., Ellis, D., Gelbart, D., Morgan, N., Peskin, B., Pfau, T., Shriberg, E., Stolcke, A., & Wooters, C. 2003. The ICSI meeting corpus. *Pages 364–367 of: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 18, 140

[71] Japkowicz, N., & Stephen, S. 2002. The Class Imbalance Problem: A Systematic Study. *Intell. Data Anal.*, **6**(5), 429–449. 68

[72] John, G. H., & Langley, P. 1995. Estimating Continuous Distributions in Bayesian Classifiers. *Pages 338–345 of: Eleventh Conference on Uncertainty in Artificial Intelligence*. San Mateo: Morgan Kaufmann. 89

[73] Juang, B., Rabiner, L., & Wilpon, J. 1987. On the use of bandpass liftering in speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **35**, 947–954. 28

[74] Kenny, P., Ouellet, P., Dehak, N., Gupta, V., & Dumouchel, P. 2008. A study of inter-speaker variability in speaker verification. *IEEE Trans. Audio, Speech and Language Processing*, 980–988. 42, 43

[75] Kenny, P., Reynolds, D., & Castaldo, F. 2010. Diarization of Telephone Conversations Using Factor Analysis. *IEEE Journal of Selected Topics in Signal Processing*, **4**(6), 1059–1070. 42

[76] Kenny, P., Stafylakis, T., Alam, J., Ouellet, P., & Kockmann, M. 2014. Joint Factor Analysis for Text-Dependent Speaker Verification. *In: Proceedings to Odyssey, 2014*. 42

[77] Kittler, J., & Hatef, M. 1998. On combining classifiers. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **20**(3), 226–239. 65

[78] Ko, A., Sabourin, R., Britto, A., & Oliveira, L. 2007. Pairwise fusion matrix for combining classifiers. *Pattern Recognition*, **40**(8), 2198–2210. 67, 72

[79] Kohavi, R. 1995. The Power of Decision Tables. *Pages 174–189 of: 8th European Conference on Machine Learning*. Springer. 89

[80] Kuncheva, L., Whitaker, C., Shipp, C., & Duin, R. 2000. Is independence good for combining classifiers? *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, **2**, 168–171. 67, 99

[81] Lam, L., & Suen, S. Y. 1997. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Trans. Systems Man and Cybernetics Part A Systems and Humans*, **27**(5), 553–568. 67, 70

[82] Li, M., Tsiartas, A., Segbroeck, M. Van, & Narayanan, S. S. 2013 (May). Speaker verification using simplified and supervised i-vector modeling. *Pages 7199–7203 of: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 44

[83] Lichman, M. 2013. *UCI Machine Learning Repository*. 14, 86

[84] López, V., Fernández, A., García, S., Palade, V., & Herrera, F. 2013. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, **250**, 113–141. 68

[85] Lu, L., Zhang, H., & Jiang, H. 2002. Content analysis for audio classification and segmentation. *IEEE Transactions on Speech and Audio Processing*, **10**(7), 504–516. 23

[86] Lu, L., Zhang, H., & Li, Z. 2003. Content-based audio classification and segmentation by using support vector machines. *Multimedia Systems*, **8**(6), 482–492. 39

[87] Luengo, I., Navas, E., Saratxaga, I., Hernáez, I., & Erro, D. 2010. AhoLab Speaker Diarisation System for Albayzin 2010. *Pages 393–396 of: FALA 2010*. 21, 140, 164, 168

[88] Luque, J., Anguera, X., Temko, A., & Hernando, J. 2008. Multimodal Technologies for Perception of Humans. Berlin, Heidelberg: Springer-Verlag. 159

[89] Macková, L., Čižmár, A., & Juhár, J. 2016 (April). Emotion recognition in i-vector space. *Pages 372–375 of: 2016 26th International Conference Radioelektronika (RADIOELEKTRONIKA)*. 55

[90] Maratea, A., Petrosino, A., & Manzo, M. 2014. Adjusted F-measure and kernel scaling for imbalanced data learning. *Information Sciences*, **257**(Apr.), 331–341. 75

[91] Markel, J., Oshika, B., & Gray, A., Jr. 1977. Long-term feature averaging for speaker recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, **25**(4), 330 – 337. 28

[92] Martin, A., Doddington, G., Kamm, T., Ordowski, M., & Przybocki, M. 1997. The DET curve in assessment of detection task performance. *In: EUROSPEECH*. ISCA. 118

[93] Meignier, S., Moraru, D., Fredouille, C., Bonastre, J., & Besacier, L. 2006. Step-by-step and integrated approaches in broadcast news speaker diarization. *Computer Speech and Language*, **20**, 303 – 330. 5, 24

[94] Meinedo, H., & Neto, J. 2003 (April). Audio segmentation, classification and clustering in a broadcast news task. *Pages II–5–8 vol.2 of: Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, vol. 2. 23

[95] Meinedo, H., & Neto, J. 2005. A stream-based audio segmentation, classification and clustering pre-processing system for broadcast news using ANN models. *In: INTERSPEECH*. 41

[96] Moattar, M.H., & Homayounpour, M.M. 2012. A review on speaker diarization systems and approaches. *Speech Communication*, **54**(10), 1065 – 1103. 23

[97] Nicholson, J., Takahashi, K., & Nakatsu, R. 1999. Emotion recognition in speech using neural networks. *Pages 495–501 vol.2 of: Neural Information Processing, 1999. Proceedings. ICONIP '99. 6th International Conference on*, vol. 2. 41

[98] NIST. *The 2009 Rich Transcription Meeting Recognition Evaluation Plan*. Available online: `http://www.itl.nist.gov/iad/mig/tests/rt/2009/docs/rt09-meeting-eval-plan-v2.pdf`. 45, 102, 110, 136

[99] NIST. *The NIST Year 2008 Speaker Recognition Evaluation Plan*. Available online: `http://www.itl.nist.gov/iad/mig/tests/sre/2008/sre08_evalplan_release4.pdf`. 117, 118

[100] Ore, B. M., Slyh, R. E., & Hansen, E. G. 2006 (June). Speaker Segmentation and Clustering using Gender Information. *Pages 1–8 of: 2006 IEEE Odyssey - The Speaker and Language Recognition Workshop*. 23

[101] Orriols-Puig, A., Bernadó-Mansilla, E., Goldberg, D. E., Sastry, K., & Lanzi, P. L. 2009. Facetwise analysis of XCS for problems with class imbalances. *IEEE Trans. Evolutionary Computation*, **13**, 1093–1119. 87

[102] Ortega, A., Castan, D., Miguel, A., & Lleida, E. *The Albayzin 2012 Audio Segmentation Evaluation*. Available online: `http://dihana.cps.unizar.es/~dcastan/wp-content/papercite-data/pdf/ortega2012.pdf`. 13, 46

[103] Paalanen, P., Kamarainen, J., Ilonen, J., & Kälviäinen, H. 2006. Feature representation and discrimination based on Gaussian mixture model probability densities—Practices and algorithms. *Pattern Recognition*, **39**(7), 1346 – 1358. 24, 31

[104] Pardo, J., Anguera, X., & Wooters, C. 2006a. *Speaker Diarization for Multi-microphone Meetings Using Only Between-Channel Differences*. Berlin, Heidelberg: Springer Berlin Heidelberg. Pages 257–264. 133, 134

[105] Pardo, J., Anguera, X., & Wooters, C. 2006b. Speaker diarization for multiple distant microphone meetings: mixing acoustic features and inter-channel time differences. *In: INTERSPEECH 2006 - ICSLP, Ninth International Conference on Spoken Language Processing, Pittsburgh, PA, USA, September 17-21, 2006*. 133

[106] Parker, J.R. 2001. Rank and response combination from confusion matrix data. *Information Fusion*, **2**(2), 113–120. 69

[107] Platt, J. 1998. Fast Training of Support Vector Machines using Sequential Minimal Optimization. *In:* Schoelkopf, B., Burges, C., & Smola, A. (eds), *Advances in Kernel Methods - Support Vector Learning*. MIT Press. 92

[108] Rabiner, L., & Juang, B. 1993. *Fundamentals of speech recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc. 27

[109] Rabiner, L. R., & Juang, B. H. 1986. An introduction to hidden Markov models. *IEEE ASSp Magazine*. 24

[110] Ranjan, S., Yu, C., Zhang, C., Kelly, F., & Hansen, J. H. L. 2016 (March). Language recognition using deep neural networks with very limited training data. *Pages 5830–5834 of: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 41

[111] Raykar, V., Yu, S., Zhao, L., Hermosillo, G., Florin, C., Bogoni, L., & Moy, L. 2010. Learning From Crowds. *Journal of Machine Learning Research*, **11**, 1297–1322. 66

[112] Read, I., & Cox, S. 2007. Automatic pitch accent prediction for text-to-speech synthesis. *Pages 482–485 of: Interspeech 2007*. 78

[113] Reynolds, D., Quatieri, T., & Dunn, R. 2000. Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing*, **10**(1), 19 – 41. 34

[114] Reynolds, D., Andrews, W., Campbell, J., Navratil, J., Peskin, B., Adami, A., Jin, Q., Klusacek, D., Abramson, J., Mihaescu, R., Godfrey, J., Jones, D., & Xiang, B. 2003. The SuperSID project: exploiting high-level information for high-accuracy. *Pages 784–787 of: Proceedings ICASSP*, vol. 4. 65

[115] Reynolds, D. A., & Torres-carrasquillo, P. 2005. Approaches and applications of audio diarization. *In: In Proc. of ICASSP*. 23

[116] Reynolds, D.A., & Torres-Carrasquillo, P. 2004. The MIT Lincoln Laboratory RT-04F diarization systems: Applications to broadcast audio and telephone conversations. *In: DARPA EARS RT-04F Workshop*. 5

[117] Rodriguez, J., Kuncheva, L., & Alonso, C. 2006. Rotation Forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(10), 1619–1630. 92

[118] Rosenberg, A. 2012. Classifying Skewed Data: Importance Weighting to Optimize Average Recall. *Pages 2242–2245 of: Interspeech 2012*. 78

[119] Rumelhart, D., Hinton, G., & Williams, R. 1985. *Learning internal representations by error propagation*. Tech. rept. DTIC Document. 92

[120] Ruta, D., & Gabrys, B. 2000. An overview of classifier fusion methods. *Computing and Information systems*, **7**, 1–10. 65, 70

[121] Rybach, D., Gollan, C., Schluter, R., & Ney, H. 2009. Audio segmentation for speech recognition using segment features. *Pages 4197–4200 of: 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 23

[122] S., Yanmin, Kamel, M.S., & Wang, Y. 2006 (Dec). Boosting for Learning Multiple Classes with Imbalanced Class Distribution. *Pages 592–602 of: Data Mining, 2006. ICDM '06. Sixth International Conference on*. 78

[123] Saedi, R., Lee, K., Kinnunen, T., Hasan, T., Fauve, B., Bousquet, P., Khoury, E., Martinez, P., Kua, J., You, C., Sun, H., Larcher, A., Rajan, P., Hautamäki, V., Hanilci, C., Braithwaite, B., Gonzalez-Hautamäki, R., Sadjadi, S., Liu, G.,

Boril, H., Shokouhi, N., Matrouf, D., El Shafey, L., Mowlaee, P., Epps, J., Thiruvaran, T., Van Leeuwen, D., Ma, B., Li, H., Hansen, J., Bonastre, J., Marcel, S., Mason, J., & Ambikairajah, E. 2013 (Aug.). I4U Submission to NIST SRE 2012: a large-scale collaborative effort for noise-robust speaker verification. *In: INTERSPEECH*. 44

[124] Schmitt, A., Schatz, B., & Minker, W. 2011. Modeling and Predicting Quality in Spoken Human-computer Interaction. *Pages 173–184 of: Proceedings of the SIGDIAL 2011 Conference*. Stroudsburg, PA, USA: Association for Computational Linguistics. 78

[125] Schuller, B., Steidl, S.n, & Batliner, A. 2009. The INTERSPEECH 2009 emotion challenge. *Pages 312–315 of: Proceedings of Interspeech 2009*. 95

[126] Schuller, B., Steidl, S., Batliner, A., Vinciarelli, A., Scherer, K., Ringeval, F., Chetouani, M., Weninger, F., Eyben, F., Marchi, E., Mortillaro, M., Salamin, H., Polychroniou, A., Valente, F., & Kim, S. 2013. The INTERSPEECH 2013 Computational Paralinguistics Challenge: Social Signals , Conflict , Emotion , Autism. *Pages 148–152 of: Proceedings of Interspeech 2013*. 78

[127] Schuller, B.W. 2012. The Computational Paralinguistics Challenge [Social Sciences]. *Signal Processing Magazine, IEEE*, **29**(4), 97–101. 65

[128] Schwarz, G. 1978. Estimating the Dimension of a Model. *Ann. Statist.*, **6**(2), 461–464. 130

[129] Sell, G., & Garcia-Romero, D. 2014 (Dec). Speaker diarization with plda i-vector scoring and unsupervised calibration. *Pages 413–417 of: Spoken Language Technology Workshop (SLT), 2014 IEEE*. 44

[130] Shah, J., Iyer, A., & Smolenski, B. 2004. Robust voiced/unvoiced classification using novel features and gaussian mixture model. *Conference on Acoustics, Speech, and Signal Processing*, **10**(May.), 10–13. 25

[131] Siegler, M., Jain, U., Raj, B., & Stern, R. 1997. Automatic Segmentation, Classification and Clustering of Broadcast News Audio. *Pages 97–99 of: Proc. DARPA Speech Recognition Workshop*. 24

[132] Silovsky, J., & Prazak, J. 2012 (March). Speaker diarization of broadcast streams using two-stage clustering based on i-vectors and cosine distance scoring. *Pages 4193–4196 of: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 44

[133] Sinha, R., Tranter, S. E., Gales, M. J. F., & Woodland, P. C. 2005. The Cambridge university march 2005 speaker diarisation system. *Pages 2437–2440 of: Interspeech*, vol. 6. 5

[134] Sjölander, K., & Beskow, J. 2000. Wavesurfer - an open source speech tool. *In: INTERSPEECH*. 19

[135] Sokolova, M., & Lapalme, G. 2009. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, **45**(4), 427–437. 77

[136] Steidl, S. 2009. *Automatic Classification of Emotion-related User States in Spontaneous Children's Speech*. Studien zur Mustererkennung. Isd. 15

[137] Tavarez, D., Navas, E., Erro, D., & Saratxaga, I. 2012a. Audio Segmentation System by Aholab for Albayzin 2012 Evaluation Campaign. *Pages 577–584 of: Proceedings of VII Jornadas en Tecnología del Habla and III Iberian SLTech Workshop (Iberspeech 2012)*. 49, 177

[138] Tavarez, D., Navas, E., Erro, D., & Saratxaga, I. 2012b. Strategies to Improve a Speaker Diarisation Tool. *Pages 4117–4121 of: LREC*. 164, 180

[139] Tavarez, D., Navas, E., Erro, D., Saratxaga, I., & Hernaez, I. 2012c. Técnicas de post-procesado de resultados en un sistema de diarización de locutores. *Procesamiento del Lenguaje Natural*, **49**(0), 109–116. 164, 180

[140] Tavarez, D., Navas, E., Erro, D., Saratxaga, I., & Hernaez, I. 2013. Nueva técnica de fusión de clasificadores aplicada a la mejora de la segmentación de audio. *Procesamiento del Lenguaje Natural*, **51**(0), 161–168. 103, 179

[141] Tavarez, D., Navas, E., Erro, D., Saratxaga, ., & Hernaez, I. 2014a. Aholab audio segmentation system for Albayzin 2014 evaluation campaign. *Pages 273–282 of: Proceedings of VIII Jornadas en Tecnologías del Habla and IV Iberian SLTech Workshop (Iberspeech 2014)*. 56, 177

[142] Tavarez, D., Navas, E., Erro, D., Saratxaga, I., & Hernaez, I. 2014b. New bilingual speech databases for audio diarization. *Pages 2666–2670 of: LREC*. 19, 21, 176

[143] Tranter, S. E. 2005 (March). Two-way cluster voting to improve speaker diarisation performance. *Pages I/753–I/756 Vol. 1 of: Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 1. 156

[144] Tranter, S. E., & Reynolds, D. A. 2006. An Overview of Automatic Speaker Diarization Systems. *IEEE Trans. on Audio, Speech and Laguage processing*, **14(5)**, 1557–1565. 1, 4

[145] van Leeuwen, D., & Huijbregts, M. 2006. *The AMI Speaker Diarization System for NIST RT06s Meeting Data*. Berlin, Heidelberg: Springer Berlin Heidelberg. Pages 371–384. 133

[146] Vandecatseye, A., Martens, J., Neto, J., Meined, H., Garcia-Mateo, C., Dieguez, J., Mihelic, F., Zibert, J., Nouza, J., David, P., Pleva, M., Cizmar, A., Papageorgiou, H., & Alexandris, C. 2004. The COST278 Pan-European Broadcast News Database. *Pages 873–876 of: 4th international conference on Language Resources and Evaluation (LREC)*. 18

[147] Vapnik, V. 1999. *The Nature of Statistical Learning Theory*. 37

[148] Ververidis, D., & Kotropoulos, C. 2006. Emotional speech recognition: Resources, features, and methods. *Speech Communication*, **48**, 1162–1181. 95

[149] Wang, C., Zou, Y., Liu, S., Shi, W., & Zheng, W. 2016 (April). An Efficient Learning Based Smartphone Playback Attack Detection Using GMM Supervector. *Pages 385–389 of: 2016 IEEE Second International Conference on Multimedia Big Data (BigMM)*. 36

[150] Wang, M., Chen, Y., Tang, Z., & Zhang, E. 2015 (Dec). I-vector based speaker gender recognition. *Pages 729–732 of: 2015 IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. 55

[151] Willsky, A. S., & Jones, H. L. 1974 (Nov). A generalized likelihood ratio approach to state estimation in linear systems subjects to abrupt changes. *Pages 846–853 of: Decision and Control including the 13th Symposium on Adaptive Processes, 1974 IEEE Conference on*. 132

[152] Woods, K., Kegelmeyer, W. P., & Bowyer, K. 1997. Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(4), 405–410. 66

[153] Xu, L., Krzyzak, A., & Suen, C.Y. 1992. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Trans. Systems, Man, and Cybernetics*, **22**(3), 418–435. 65, 67, 71, 96

[154] Yang, Q., & Wu, X. 2006. 10 Challenging Problems in Data Mining Research. *IEEE Int. Conf. Data Mining*, **5**(4), 597–604. 67

[155] Young, J., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., & Woodland, P. 2006. *The HTK Book Version 3.4*. Cambridge University Press. 29

[156] Young, S. 1996. A review of large-vocabulary continuous-speech. *Signal Processing Magazine, IEEE*, **13**(5), 45. 4, 25, 30

[157] Young, S.J. 1994. The HTK Hidden Markov Model Toolkit: Design and Philosophy. *Entropic Cambridge Research Laboratory, Ltd*, **2**, 2–44. 58

[158] Zelenák, M., Schulz, H., & Hernando, J. 2012. Speaker diarization of broadcast news in Albayzin 2010 evaluation campaign. *EURASIP Journal on Audio, Speech, and Music Processing*, 1–9. 12, 18, 46, 164

[159] Zelenák, M., & Hernando, J. 2010. On the improvement of speaker diarization by detecting overlapped speech. *Pages 153–156 of: Actas de las VI jornadas en tecnologías del habla*. 159

[160] Zhang, S., Yang, M., Cour, T., Yu, K., & Metaxas, D. N. 2015. Query Specific Rank Fusion for Image Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37**(4), 803–815. 66

[161] Zhou, B., & Hansen, J. 2000. Unsupervised Audio Stream Segmentation And Clustering Via The Bayesian Information Criterion. *Pages 714–717 of: ISCLP 2000*. 5, 24

[162] Zhu, X., Barras, C., Meignier, S., & Gauvain, J.-L. 2005 (sept 2005). Combining speaker identification and bic for speaker diarization. *In: Interspeech'05, ISCA*. 157, 160

Declaration

I herewith declare that I have produced this work without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such. This work has not previously been presented in identical or similar form to any examination board.

The dissertation work was conducted from 2012 to 2016 under the supervision of Eva Navas Cordón at the University of the Basque Country.

Bilbao, October 2016,

David Tavárez Arriba.

This dissertation was finished writing in Bilbao on Monday 24th October, 2016