Caracterización analítica de segmentos del habla mediante clustering de vectores de coeficientes cepstrales

Ingeniería en Informática: Proyecto Fin de Carrera

Universidad Carlos III de Madrid

Escuela Politécnica Superior

Rafael Gálvez Vizcaíno

Tutor: Luis Puente Rodríguez

Madrid - 30 de octubre de 2014

Agradecimientos

Gracias a todos los que me han ayudado a acabar este largo camino, en especial a mi madre y a Stephanie por acompañarme en cada momento y a Dani y a Luis por ser mis referentes en la universidad.

Gracias a toda la gente de Guadalupe: a los vikingos, a todos los niños y API de Posco y de Génesis y a los MsPS (especialmente a Giancarlo por su cercanía, su manera de trabajar y por la sintonía que compartimos).

Gracias también a mis primeros compañeros de trabajo, tanto de CESyA como de LEAP, con mención especial a José porque me ayudó a disfrutar del trabajo en el CESyA, a Luis por confiar en mí tanto laboralmente como académicamente y ayudarme con su innata pasión por enseñar a concretar este PFC; y a Azul, a Elijah y a Mcnair por mostrarme la belleza de trabajar desarrollando Software Libre que aporta valores necesarios en la sociedad actual.

De la universidad, gracias a Jesús por ser mi compañero desde principio hasta el final, a Míguel por ser un referente de inquietudes profesionales, y a Míriam por su paciencia conmigo en algunas prácticas y su esfuerzo por seguir juntándonos y viéndonos los cuatro al empezar nuestras vidas profesionales.

Por último, gracias a Dios por regalarme cada día de mi vida, la pasión por la realidad que me rodea y el Evangelio que me mueve a actuar. Sin Él nada de esto tendría sentido.

KEY WORDS

Speech recognition, clustering, unsupervised classification, speech fragments, mfcc, machine learning.

ABSTRACT

This project belongs to a greater research which aims to identify a correspondence between speech fragments and regions from their feature vector space. If it were to be true, the GMM models used during the conventional speech recognition scheme could be replaced by discrete and deterministic units, helping to reduce the dimensionality of the problem solved by HMM.

This Final Degree Project is the first step towards that goal. It will propose a clustering algorithm that uses the cepstral coefficients to characterize speech fragments. During its development, a series of experiments will be carried out by an unsupervised classification algorithm named Subtractive Clustering, which avoids the need of specifying some kind of model for those clusters there is no knowledge about.

PALABRAS CLAVE

Reconocimiento del habla, clustering, clasificación no supervisada, segmentos del habla, MFCC, aprendizaje automático.

RESUMEN

Este proyecto se enmarca dentro de una investigación mayor cuyo objetivo es establecer si distintos fragmentos de habla ocupan distintas regiones propias e independientes del espacio de vectores de características. En caso afirmativo, podrán sustituirse los modelos GMM de los reconocedores del habla convencionales por unidades discretas y deterministas que ayuden a reducir la dimensionalidad del problema afrontado por HMM.

El primer paso de esa investigación constituye el contenido de este Proyecto Fin de Carrera, cuyo cometido es proponer un algoritmo de clustering que establezca la posibilidad de realizar una estudio analítico del habla a través de coeficientes cepstrales extraídos de fragmentos en castellano. Durante su desarrollo se realizarán pruebas que determinen la existencia de conjuntos de vectores de características, de manera que se pueda establecer una relación entre ellos y los bifonemas a los que corresponden sus vectores. Con este fin se utilizará el algoritmo de clasificación no supervisada Subtractive Clustering, que permitirá evitar la dificultad que supone el desconocimiento de la forma que esos conjuntos adoptan.

Contenido

Índice de ilustraciones

Índice de tablas

1 Introducción

El presente proyecto pertenece al dominio del procesamiento del habla, el cual comprende tres áreas diferentes: la síntesis de voz, el reconocimiento de locutor y el reconocimiento de habla.

La síntesis de voz tiene como objetivo la generación automática de habla a partir de su representación simbólica. Constituye tanto una aplicación informática útil para acceder mediante el habla a información almacenada en forma escrita, como una herramienta de investigación en fonética que permite validar hipótesis sobre producción y percepción del habla realizadas desde diversos marcos teóricos (Llisterri et al., 1999).

Por otra parte, el reconocimiento de locutor se puede definir como el proceso automático por el que se determina quién está hablando basándose en la información contenida en la señal acústica producida. La naturaleza ha dotado al ser humano de la capacidad de reconocer a las personas por su voz, ya que ésta transporta, junto a la información semántica, datos relativos a la identidad del emisor (Puente, 2014).

Por último, el reconocimiento de habla busca generar una transcripción escrita a partir de una señal de audio (Li & Sim, 2014), donde uno de los principales problemas es la variabilidad intrínseca de la señal producida por la voz, procedente de las características anatómicas (Benzeghiba et al., 2007), sociales, culturales de la persona que la emite, así como de las características propias de la locución presente en ella, tales como la realización concreta de los fonemas y su coarticulación en palabras y frases (Puente, 2014).

Actualmente, esta variabilidad no está lo suficientemente acotada y eso supone un obstáculo para la implementación de sistemas de reconocimiento automático de habla. Un sistema puntero como el construido por Google, con un amplio vocabulario e independiente de locutor, obtiene resultados como los mostrados en (Cucu et al. 2014) y en (Bacchiani et al., 2014), los cuales muestran que el estado del arte del reconocimiento automático del habla tiene problemas de precisión cuando el diccionario manejado

contempla un gran número de palabras y su objetivo contempla la independencia de locutor.

El modo en que los sonidos son representados en la mente de las personas es importante para modelar dicha variabilidad (De Lacy, 2007). Esta relación es estudiada por la fonología ayudándose de la descripción que hace la fonética de los sonidos del habla (Ladefoged & Johnson, 2014).

La tecnología actual persigue obtener resultados de reconocimiento comparables a los logrados por los seres humanos; para ello se propone un esquema estándar para un sistema de reconocimiento automático de habla compuesto por un extractor de características, un modelo acústico, un léxicon y un modelo de lenguaje (Li & Sim, 2014).

El problema que se quiere afrontar, y de cuya solución este proyecto forma parte, es responder a la pregunta de si distintos fragmentos de habla ocupan distintas regiones propias e independientes de un espacio definido por ciertas características presentes en ellos. A este proyecto se le ha requerido proponer un algoritmo de clusterización que identifique las agrupaciones de vectores (o burbujas, según la terminología propia de este proyecto) en su espacio multidimensional, de modo que sirva de apoyo a la investigación que permita abordar la pregunta mencionada.

1.1 Motivación

Las técnicas de inteligencia artificial y minería de datos han sido ampliamente utilizadas para resolver problemas de clasificación, diagnóstico, computación, predicción y optimización en multitud de dominios (Xie, Fan, Wang, Lu, & Li, 2014). El clustering ha recibido recientemente mucha atención por su utilidad para resolver problemas de tipo analítico (Ayyasamy, 2013), y dada su naturaleza no supervisada (Jain, Murty, & Flynn, Data clustering: a review, 1999) puede constituir una vía de solución alternativa a las técnicas estadísticas más utilizadas actualmente para el problema abordado.

Si bien existen propuestas como (Hinton et al., 2012), que para resolver el problema de la

caracterización de segmentos de habla durante la fase de reconocimiento utiliza técnicas estadísticas de aprendizaje (en concreto, Redes Neuronales), la aproximación que este proyecto realiza es diferente por su naturaleza analítica.

La motivación que impulsa la ejecución de este trabajo es encontrar una vía para responder a la pregunta de si es posible establecer una relación unívoca entre segmentos de habla y regiones concretas del espacio de vectores de características, y cuál es la estabilidad de esas regiones en relación al locutor y a las causas de la variabilidad de la producción de la voz.

Si la pregunta planteada obtuviera una respuesta afirmativa, se podrían realizar ulteriores investigaciones para desvelar el rol de esta correspondencia en la fase de reconocimiento de habla, basándose en las técnicas y las conclusiones obtenidas durante su resolución.

Por el contrario, en caso de que la respuesta fuera negativa, se podrían buscar otras técnicas que, ayudándose de los resultados y las conclusiones de este trabajo, permitieran abordar el problema desde un punto de vista diferente.

En cualquier caso, el ejercicio de desarrollo y experimentación será útil para el conocimiento humano relativo al área del procesamiento del habla, ayudando a que posteriores trabajos innoven tanto en las técnicas como en los supuestos a la hora de afrontar problemas similares a este.

1.2 Propósito

El propósito de este trabajo no es otro que desarrollar una algorítmica y un proceso de prueba y análisis que validen la resolución final de la respuesta.

Para la construcción de la algorítmica de identificación de las regiones de vectores se emplearán elementos presentes en el estado del arte o, en caso de ser necesario, planteados expresamente para el objetivo propuesto. El proceso resultante ha de ser de naturaleza determinista, en oposición a las técnicas estocásticas más populares.

Las conclusiones deben tener en cuenta las limitaciones impuestas por el alcance de la

respuesta obtenida, el volumen de datos manejados y las características de la algorítmica (como la imposibilidad para probar infinitos parámetros en las entradas).

1.3 Presentación de la estructura del documento

En la sección actual se ha realizado la presentación del proyecto y de su contexto, motivación y objetivo. El resto del trabajo se estructura de la siguiente forma: en la sección 2 se profundizará en el Estado del arte, en la sección 3 se dará forma al planteamiento propuesto para que, en la sección 4, se describa el desarrollo experimental. En la sección 5 se mostrarán los resultados, en la sección 6 se presentarán las conclusiones y los trabajos futuros, en la sección 7 el presupuesto, y finalmente las referencias consultadas. En el apéndice A se adjuntan los datos relativos a una de las gráficas de los resultados.

2 Estado del arte

2.1 Percepción del habla humana

El mejor sistema de reconocimiento de habla en la naturaleza es el oído humano. Para poder emular su comportamiento, es necesario tener en cuenta que, por razones de coevolución, los sistemas auditivo y fonador se han desarrollado paralelamente y por lo tanto los mecanismos de producción del habla están íntimamente vinculados a los de su percepción; no es posible entender cómo se reconoce sin entender cómo se produce (Puente, 2014).

Por otra parte, la forma en que los sonidos son representados en la mente de las personas es un factor clave en el reconocimiento de los sonidos del habla, ya que el mensaje es entendido por el cerebro a partir de las funciones neurológicas que la soportan (Zatorre, 2005).

2.1.1 Producción de la voz

La fonación

La voz se produce gracias a la acción coordinada de casi todo nuestro cuerpo. El aparato fonador está integrado por estructuras musculares de diferentes regiones y por elementos del aparato respiratorio y del aparato digestivo (Torres, Antonio, & Casado, 2007).

En el periodo previo a la producción del sonido, el diafragma se contrae y los pulmones recogen aire por las vías respiratorias. Inmediatamente antes de la fonación, los pliegues vocales mantienen cerrada la glotis impidiendo la circulación del aire. A continuación, la musculatura abdominal se contrae mientras el diafragma se relaja y aumenta la presión intraalveolar, la cual se transmite por la tráquea hasta la región subglótica. Cuando esta presión supera a la fuerza de cierre de la glotis, las cuerdas vocales son obligadas a separarse y el aire sale con fuerza, produciéndose un descenso brusco de la presión

subglótica. La reducción de esta presión junto con la elasticidad de los pliegues vocales produce que la glotis vuelva a cerrarse y el ciclo comienza de nuevo (Torres, Antonio, & Casado, 2007).

Este proceso se repite en rápida secuencia durante toda la fonación, produciendo un tren de ondas de presión que constituyen el componente fundamental de la voz vocálica. Todas las cavidades situadas por encima de las cuerdas vocales actúan como cajas de resonancia de ella (boca, faringe y fosas nasales). Hay resonadores móviles, como la boca, que pueden modificar su forma y volumen adaptándose al sonido producido et al. fijos, como las fosas nasales, que no podrán cambiar su forma ni su volumen (Puente, 2014).

Articulación de sonidos

El flujo proveniente de la laringe es alterado por la posición concreta de los órganos articulatorios (lengua, paladar, dientes, labios) (Puente, 2014).

En la producción de los sonidos, alguno de esos órganos, denominado articulador activo (la punta de la lengua, labio inferior, dorso de la lengua...) se ubica respecto a un articulador pasivo (labio superior, incisivos superiores, alveolos, paladar duro, velo palatino) para interponer un obstáculo al flujo de aire o formar un estrechamiento en el conducto vocal (Hualde, 2005).

Los sonidos emitidos tienen características relevantes como su frecuencia fundamental, su intensidad y su timbre, y también dependen de sus coarticulaciones con los anteriores y los siguientes sonidos (Torres, Antonio, & Casado, 2007).

Frecuencia fundamental e intensidad

La frecuencia del tren de ondas de presión se denomina frecuencia fundamental, está presente en los sonidos vocálicos producidos por la vibración de las cuerdas vocales ante la presión del aire en el ciclo fonatorio y está directamente relacionada con su grosor y su longitud (Torres, Antonio, & Casado, 2007).

Por otro lado la intensidad de la voz dependerá principalmente de la presión del aire

espirado. La energía con la que el aire es impulsado desde los pulmones determinará una mayor o menor amplitud vibratoria de los pliegues vocales, que provocará un aumento o disminución de la intensidad del sonido producido (Torres, Antonio, & Casado, 2007).

Timbre

El sonido producido en los pliegues vocales es un tono complejo, que consta de una frecuencia fundamental y de los armónicos que la acompañan, alterado todo ello por las reflexiones en los resonadores que desde este punto de vista constituyen el tracto vocal. La onda formada en la laringe pasa a través de las cavidades supraglóticas que actúan como filtros, realizando aquellas frecuencias que se sitúan en torno a las frecuencias propias de las cavidades de resonancia y atenuando las demás. El conjunto formado por el tono fundamental más los armónicos modificados constituyen el timbre de la voz (Torres, Antonio, & Casado, 2007).

Coarticulación

En lingüística y especialmente en fonética y fonología, se considera como segmento cualquier unidad discreta que pueda ser identificada emisiva o auditivamente en el flujo de producción del habla. El segmento es un concepto muy cercano al de fonema, y se considera, desde este punto de vista analítico, discreto porque puede ser identificado individualmente (Puente, 2014).

En la producción de habla, antes de haberse establecido las posiciones finales relativas de los órganos del tracto vocal que constituyen una articulación, los órganos ya van tomando posiciones para realizar la siguiente, de forma que algunas de las características del siguiente sonido se anticipan en el actual (Planas, 2012).

Es por ello que, desde un punto de vista fisiológico o de teoría de la señal, la articulación de sonidos no es discreta; por el contrario, pasar de la emisión de un fonema a otro es un proceso continuo en el que los órganos articulatorios fluyen de una posición a la siguiente sin dejar de producir sonido. La mezcla de las características de dos sonidos consecutivos

(o incluso de otros más alejados), es lo que se conoce como coarticulación, y el periodo difuso, sin claros límites, que lleva de un estado de fonación al siguiente es el denominado periodo coarticulatorio (Puente, 2014).

2.1.2 El oído humano

Los sistemas de reconocimiento automático de habla tratan de emular el funcionamiento del oído, un complejo sistema que está dividido, desde un punto de vista biológico, en 3 secciones: oído externo, oído medio y oído interno (Puente, 2014).

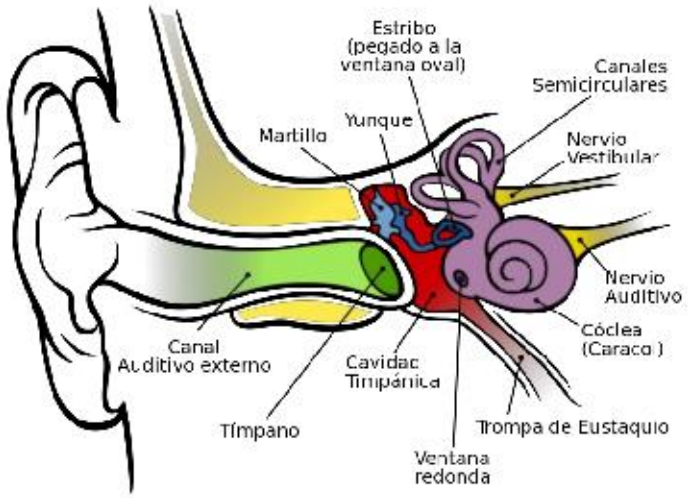

Ilustración 1: Sistema auditivo humano

El mecanismo de percepción del sonido comienza en el oído externo con la captación de las ondas de presión, que son dirigidas por el canal auditivo hasta el tímpano, el cual vibra por la acción de los cambios temporales de la presión del aire sobre él (Puente, 2014).

A lo largo del oído medio, esta vibración es transmitida por tres pequeños huesos, martillo, yunque y estribo, que realizan la adaptación de las impedancias del oído externo con las del oído interno, convirtiendo el movimiento de baja presión y alta amplitud en el tímpano, en otro de alta presión y baja amplitud en el apoyo del estribo sobre la cóclea denominado ventana oval (Puente, 2014).

Finalmente, en el oído interno las vibraciones de la ventana oval son caracterizadas por la

membrana basilar y convertidas en señales eléctricas por el órgano de Corti, las cuales son enviadas al cerebro. La membrana basilar posee tres propiedades que son básicas en este proceso: su respuesta en frecuencia no es uniforme, en potencia es logarítmica, y no distingue frecuencias muy cercanas sino más bien grupos de frecuencias (Puente, 2014).

2.1.3 Fonología y fonética

La fonética es la ciencia que estudia los sonidos en su realización concreta, y la fonología la disciplina que estudia los sonidos del lenguaje en tanto que elementos funcionales en un sistema de comunicación lingüística (Obediente, 1998).

Toda interacción lingüística entre personas presupone la existencia previa de un sistema compuesto de un número limitado de elementos diferenciados por caracteres muy precisos. Las unidades utilizadas como signos en la lengua hablada son sonidos y agrupaciones de sonidos, que deben estar diferenciadas de tal forma que el oído humano pueda, sin equivocarse, identificar e interpretar las diferencias entre ellos, así como el aparato fonador pueda producirlos de manera reconocible (Etxebarria, 2013).

Dentro del sistema fonológico de una lengua, se define como fonema la unidad menor distinguible en él. Cada uno de ellos es realizado de manera diferente en función del resto de fonemas de su entorno y de las características propias del locutor, entre las que se encuentran el acento propio de la persona y del dialecto, la longitud de su tracto vocal, o el volumen (Lee, Hayamizu, Hon, Huang, Swartz, & Weide, 1990). Para definir estas variaciones se utiliza el concepto de alófono, el cual da nombre a la realización de un fonema en un entorno particular (Puente, 2014) y que es el elemento utilizado por las personas para descomponer los sonidos en segmentos de palabras y palabras completas (Mitterer, Scharenborg & McQueen, 2013).

2.2 Reconocedores automáticos de habla

Como se afirmó en la sección 1, el objetivo de un reconocedor automático de habla es obtener una transcripción escrita a partir de una señal de audio que constituye su entrada.

Para resolver este problema, existe un esquema estándar para el sistema de reconocimiento compuesto por un extractor de características, un modelo acústico, un léxicon y un modelo de lenguaje. Este esquema es reflejado por la Ilustración 2.

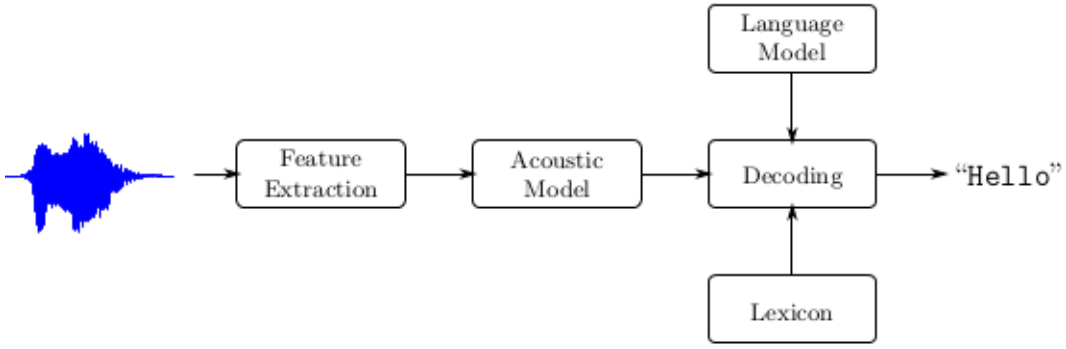

Ilustración 2: Arquitectura de un sistema de reconocimiento de habla

En primer lugar, la extracción de características (feature extraction) preprocesa y transforma la señal de habla en un conjunto de características que se agrupan en vectores y que mantienen únicamente la información lingüística considerada relevante. Estos vectores constituyen la entrada del modelo acústico (acoustic model), que los comparará con las representaciones obtenidas previamente durante su proceso de entrenamiento (ver apartado 2.2.1) para obtener una serie de símbolos que serán evaluados a través del modelo de lenguaje y del diccionario o léxicon. De esta manera, se eligen aquellos que tengan mayor probabilidad de constituir una unidad semánticamente significativa (Li & Sim, 2014).

2.2.1 Fases de reconocimiento

Como sistema de reconocimiento biométrico, la implementación de un reconocedor de habla se compone de dos fases: fase de entrenamiento y fase de reconocimiento (Puente, 2014).

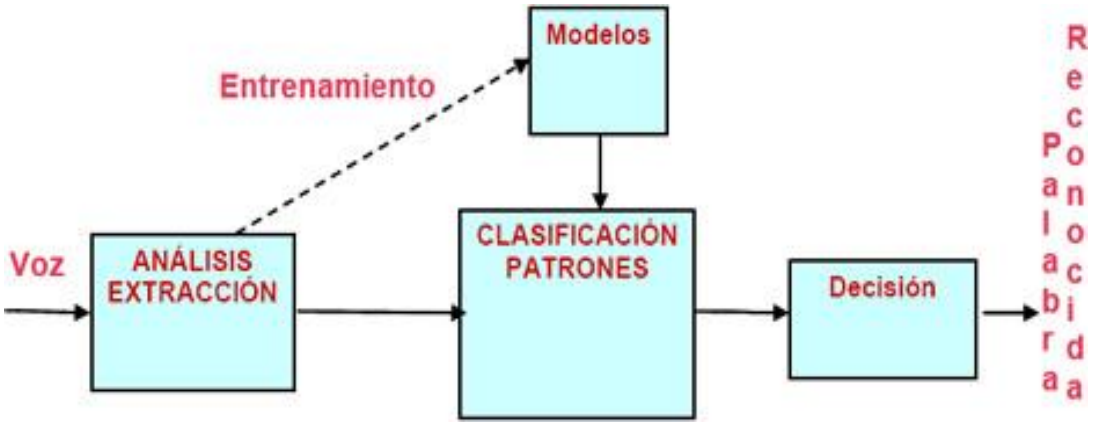

Ilustración 3: Fases de un sistema de reconocimiento de habla

Durante el entrenamiento, se determinan los valores de los parámetros que definen el modelo acústico utilizando un conjunto de datos que permita establecer una correspondencia entre las características de entrada y los símbolos de salida (Solera, 2011).

En la fase de reconocimiento, estos modelos se utilizan para clasificar los vectores de características en base a los parámetros obtenidos en la fase de entrenamiento, de manera que se emitan los símbolos necesarios para tomar la decisión final de reconocimiento.

Extracción de características

La representación acústica de una señal de voz contiene mucha información difícil de ser procesada en su forma de original debido a la gran cantidad de elementos no relacionados con su lingüística que posee. Para evitar su consideración, la fase de extracción de características produce vectores numéricos cuyos valores son calculados de tal manera que los elementos innecesarios de la señal son descartados y que reflejan aquellas propiedades que resultan de interés para el sistema de reconocimiento de habla (Li & Sim, 2014).

Debido a que estas propiedades varían con el tiempo, es necesario realizar un análisis en pequeños intervalos en los cuales se asume que las características estadísticas de la señal

permanecen aproximadamente. De esta forma se obtienen vectores de características suficientemente representativos de los segmentos cuasi-estacionarios de la señal de entrada (tramas) (Solera, 2011).

El resultado de este proceso es una secuencia de vectores de características que contiene información acerca de la evolución temporal de la señal de voz (Solera, 2011).

En la actualidad se emplean diversos tipos de características que presentan distintas propiedades. En general, incorporan en su análisis algún aspecto relacionado con la fonación y con la percepción auditiva (Solera, 2011).

Por tener un uso más extendido, se pueden destacar 3 tipos: MFCC (Mel-Frequency Cepstral Coefficients), LPC (Linear Prediction Coefficients) y PLP (Perceptual Linear Coefficients) (Solera, 2011). A continuación se describirán cada uno de ellos, así como PNCC (Power-Normalized Cepstral Coefficients), basado en el primero.

LPC

Linear Prediction Coefficients se basa en que las muestras consecutivas presentes en un audio tienen una alta correlación entre ellas. Por ello, propone expresar cada muestra de la señal como una combinación lineal de las anteriores (Chauhan, Soni, & Zafar, 2013).

Este tipo de características produce estimaciones de la intensidad y de la frecuencia eliminando previamente los formantes (ver sección Timbre) de la señal de audio a través de un filtrado inverso, lo cual disminuye la influencia de las características propias del locutor en la caracterización de la señal. Los formantes pueden ser estimados con suficiente precisión gracias a que el tracto vocal puede ser modelado y analizado con gran precisión (Chauhan, Soni, & Zafar, 2013).

Dadas sus propiedades, LPC también se utiliza para identificar locutores (Wu & Lin, 2009), y constituye un punto de partida para la propuesta de nuevos tipos de características como PLP (Hermansky, 1990).

PLP

Perceptual Linear Coefficients (Hermansky, 1990) mejora las prestaciones de LPC reduciendo la sensibilidad de las características de la señal de voz a posibles distorsiones, empleando técnicas sobre el espectro que identifican las bandas críticas, simulan el incremento de la sensibilidad auditiva con la frecuencia y tratan de reproducir la intensidad del sonido y la percibida (Moreno, 2002).

Si bien tanto LPC como PLP eliminan la información contenida en los formantes, este último refleja la información lingüística de un mensaje sin depender tanto de la forma del espectro de la señal. Esta mejora es refrendada por experimentos llevados a cabo con varias bases de datos de audio y diferentes idiomas, lo que unido a su mayor eficiencia computacional lo establece como una de las principales alternativas a LPC (Hermansky, 1990).

PNCC

Power-Normalized Cepstral Coefficients es un tipo de características que trabaja con filtros Gammatone (Valero & Alías, 2012) en vez de utilizar la escala de Mel (como MFCC) (Kim & Stern, Power-normalized cepstral coefficients (PNCC) for robust speech recognition, 2012).

Igual que LPC, este tipo de características se ha utilizado en investigaciones relacionadas con el reconocimiento de habla (Kim & Stern, Power-normalized cepstral coefficients (PNCC) for robust speech recognition, 2012) y también para el reconocimiento de locutor (Ambikairajah, Kua, Sethu, & Li, 2012).

MFCC

Mel-Frequency Cepstral Coefficients incorpora consideraciones perceptuales que emulan el funcionamiento del oído interno (Solera, 2011). Basándose en las tres propiedades analizadas en la sección El oído humano, los coeficientes cepstrales tratan de modelar la información relevante de la señal acústica de forma que se imite el procesamiento que

éste hace de forma natural (Chauhan, Soni, & Zafar, 2013).

Los coeficientes MFCC y el logaritmo de la energía de cada una de las tramas de la señal se unen en un único vector de coeficientes estáticos o de primer orden, a los cuales se les pueden adicionar sus componentes dinámicas (Solera, 2011).

La parametrización convencional utiliza 12 coeficientes de estáticos, el coeficiente de la energía y los correspondientes parámetros dinámicos (velocidades o primeras derivadas, aceleraciones o segundas derivadas).

Es práctica habitual aplicar sobre estos vectores de características algún tipo de normalización, con el fin de eliminar diversos efectos asociados al ruido y al canal de comunicaciones (Solera, 2011).

Modelo acústico

La etapa de modelado acústico tiene por objetivo proporcionar la información de nivel fonético (ver sección 2.3) necesaria en el proceso de decodificación, de manera que se puedan relacionar los vectores de características de la entrada con los símbolos utilizados durante el entrenamiento del modelo.

Para establecer esta relación, y partiendo del supuesto de que la señal de voz se puede caracterizar de forma apropiada como un proceso estocástico paramétrico cuyas variables pueden ser estimadas con suficiente precisión (Li & Sim, 2014), el estado del arte utiliza modelos ocultos de Markov (HMM),

El problema del reconocimiento del habla puede modelarse matemáticamente de la siguiente manera:

> *dada una secuencia de observaciones acústicas, el objetivo del reconocimiento automático de habla es encontrar la secuencia de palabras $W^{\star}$ que maximiza la probabilidad de que*
>
> $$W^{*} = argmax\_W[p(W|O)] = argmax\_W\ [p(O|W)p(W)]$$

donde W representa las palabras a reconocer, W todas las posibles palabras y O la señal de habla observada* (Li & Sim, 2014).

La tarea de seleccionar la unidad acústica básica es un aspecto fundamental en el diseño de los reconocedores de habla, ya que tendrán influencia en el número de estados de los HMM. Las unidades que mejor capturan el efecto de la coarticulación (ver sección 2.1.1) son aquellas que dependen de su contexto (como los fonemas, bifonemas, trifonemas (Solera, 2011).

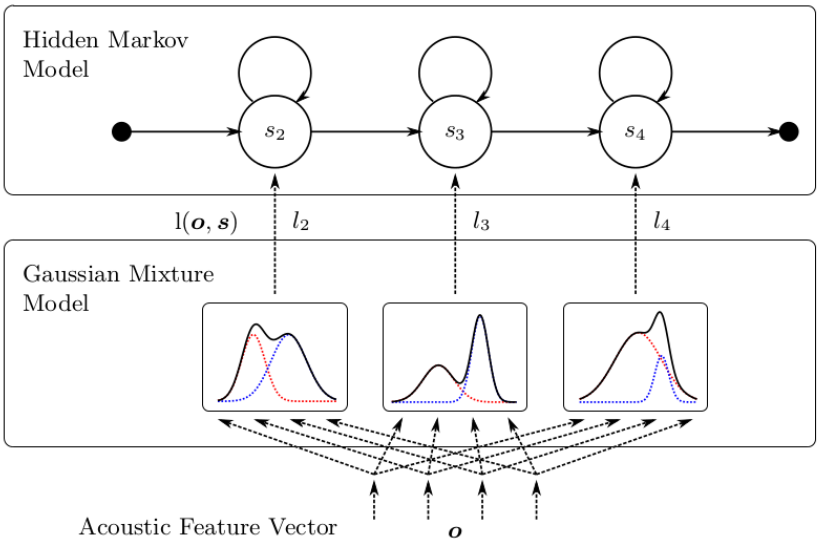

Ilustración 4: Relación entre GMM y HMM

En los reconocedores convencionales, como se muestra en la Ilustración 4, para cada estado del HMM se utiliza un GMM que representa la distribución de probabilidad de todas las características de ese estado (Li & Sim, 2014). Así, se puede afirmar que en reconocedores GMM/HMM las distribuciones de probabilidad de emisión las modelan GMM (Solera, 2011).

Esta binomio, por contra, ofrece una pobre capacidad discriminativa entre distintas clases acústicas debido, entre otros motivos, a la influencia que tienen las distorsiones de las distribuciones de probabilidad de los parámetros espectrales sobre los modelos de mezclas de Gaussianas (Solera, 2011).

Modelo de lenguaje

El modelo de lenguaje es un conjunto de reglas que relacionan unidades acústicas básicas (fonemas, bifonemas, trifonemas…) con palabras o frases. Estas reglas de combinación se fijan habitualmente mediante diccionarios, que se usarán posteriormente para construir los modelos acústicos de las palabras mediante la concatenación de los HMM de las clases acústicas que las forman (Solera, 2011).

Asimismo, el nivel de análisis sintáctico establece qué combinación de palabras son válidas para formar frases sintácticamente correctas. En los casos más sencillos, con un vocabulario reducido, se usan gramáticas deterministas (reglas), pero este procedimiento no es viable en el caso de tareas más complejas, por lo que se emplean modelos probabilísticos del lenguaje (habitualmente n-gramas) obtenidos mediante el análisis estadístico de grandes bases de datos de textos (Solera, 2011).

Léxicon

El léxicon define una relación unívoca entre una palabra y la secuencia de unidades acústicas que la modela (Li & Sim, 2014).

También determina el vocabulario del sistema de reconocimiento automático de habla, que se define como el conjunto de todas las posibles palabras que el sistema puede reconocer. Su tamaño tiene un impacto directo sobre el rendimiento del sistema: incrementándolo, el espacio de probabilístico búsqueda crece y con ella la complejidad del algoritmo que lo recorre (Li & Sim, 2014).

2.2.2 Sistemas software de reconocimiento de habla

Existen paquetes de software que implementan sistemas de reconocimiento de habla utilizando las fases descritas en la sección 2.2.1. Resulta de interés analizar las arquitecturas software que implementan este tipo de sistemas, cómo se utilizan toolkits que permiten experimentar con ellos, y los resultados que desde un punto de vista

puramente utilitario están obteniendo implementaciones comerciales.

En esta sección se introducirá el estado del arte de este tipo de proyectos software. Debido a que cada uno es gobernado por una licencia específica que limita la información que se puede obtener de ellos, sus análisis serán de naturalezas diferentes.

HTK

HTK (Young et al., 1997) es un toolkit para construir Hidden Markov Models (HMM) que contiene herramientas capaces de generar tanto extractores de características como clasificadores.

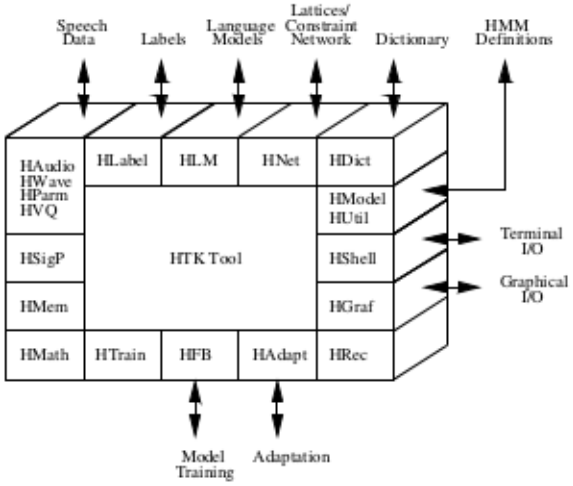

Ilustración 5: Arquitectura Software de HTK

HTK proporciona cuatro herramientas básicas para la estimación de parámetros de modelos HMM. Dos son utilizadas para la inicialización, y otras dos para la reestimación de los parámetros. Esta implementación permite experimentar fácilmente eligiendo qué utilidades se emplean, y cuáles no.

HTK distingue dos tipos de modelos HMM: de palabras completa y de parte de palabra (típicamente fonemas). Dependiendo del tipo escogido, se modelará una u otra unidad. Los modelos HMM de palabra completa suelen ser entrenados con muestras de cada palabra aislada.

Las herramientas de HTK están orientadas principalmente a la construcción de modelos HMM de parte de palabras en las que las unidades básicas son los fonemas. Para cada fonema se construye un modelo HMM y el habla continua es procesada uniendo los fonemas de acuerdo a un diccionario de pronunciación.

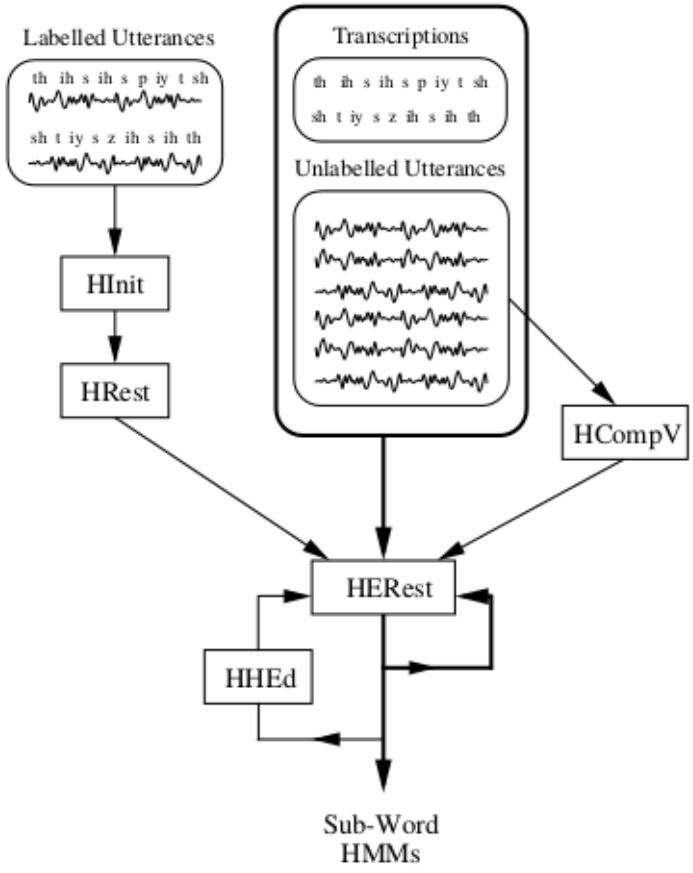

Ilustración 6: Entrenamiento de modelos HMM de parte de palabra

La inicialización de los modelos HMM de partes de palabra, como se observa en la Ilustración 6, puede realizarse de dos maneras distintas: utilizando la información provista en el etiquetado, o realizando una estimación de las ubicaciones de los fonemas en el audio de entrenamiento.

Sphinx 4

Sphinx 4 (Walker et al., 2004), creado por la Universidad de Carnegie Mellon (CMU), está

orientado a su utilización como librería y por ello su arquitectura software ha sido diseñada con el objetivo de ser modular y muy flexible.

Sphinx ofrece 3 módulos que implementan las fases de reconocimiento descritas en el apartado 2.2.1: el front-end, el decodificador y el lingüista (ver Ilustración 7).

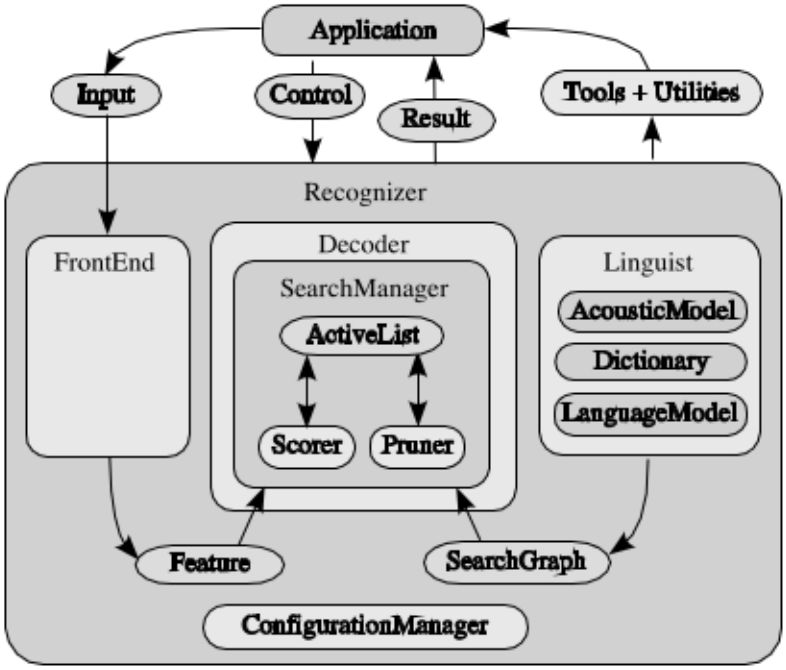

Ilustración 7: Arquitectura de Sphinx 4

El front-end recibe como entrada una o varias señales de audio y las parametriza en una secuencia de características, produciendo típicamente MFCC. El diseño de Sphinx es lo suficientemente flexible como para poder cambiar de algoritmo en el front-end sin necesidad de realizar modificaciones en el interior del sistema. Debido a que también implementa la generación de coeficientes PLP, es posible utilizar éstos en vez de los MFCC fácilmente.

El lingüista está compuesto por 3 elementos: el modelo de lenguaje, el diccionario y el modelo acústico. A través de ellos genera un grafo de búsqueda que el decodificador utilizará para obtener un resultado de reconocimiento.

Éste, por su parte, tiene como función principal generar hipótesis de resultados a partir de

las características del front-end y el grafo de búsqueda del lingüista. Un gestor de búsquedas reconocerá un conjunto de características y creará resultados a ser procesados con utilidades de Sphinx capaces de producir una matriz de resultados y de puntuaciones.

Dragon NaturallySpeaking

La empresa Nuance ofrece un paquete de reconocimiento automático de habla llamado Dragon NaturallySpeaking (DNS), el cual en el momento de redacción de esta memoria se encuentra en su versión 13.

Entre sus características más importantes, pueden destacarse su capacidad de aprendizaje, su integración con otros paquetes software de uso común como procesadores de texto y navegadores web, y su facilidad de uso desde el primer momento.

DNS es capaz de aprender las palabras y las frases que el usuario más utiliza, incluyendo nombres propios, y también permite crear listas de comandos y palabras personalizados.

Asimismo, es capaz de interaccionar con procesadores de textos (ver Ilustración 8), herramientas de generación de presentaciones, hojas de cálculo... así como con navegadores web, simplificando la realización de búsquedas y la visita de sitios web.

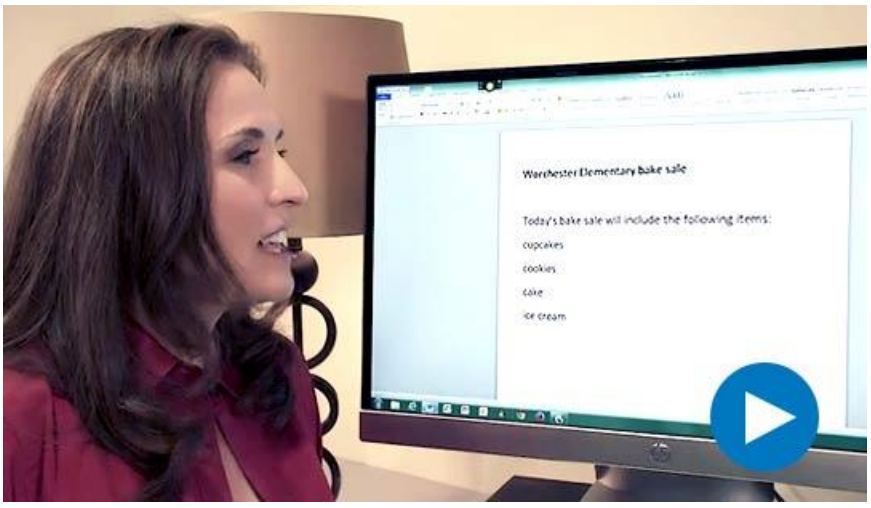

Ilustración 8: Mujer utilizando Dragon NaturallySpeak

iSpeech

iSpeech es un proveedor de reconocimiento y síntesis de habla.

El servicio de reconocimiento de habla recibe audio ya sea desde un micrófono o a través un fichero de audio, y entrega su transcripción. Se dispone de diferentes modelos de lenguaje que permiten que el usuario/desarrollador escoja aquel que mejor se adapta a las características buscadas: a menor cantidad de palabras contempladas, más rápido y preciso será el reconocimiento.

iSpeech ofrece su SDK (Software Development Kit), con licencia Open Source, para utilizar su API (Application Programming Interface) desde aplicaciones móviles (Android, iPhone, Blackberry) y desde páginas web.

Web Speech

Web Speech define una API para el lenguaje Javascript que permite a desarrolladores web incorporar tanto síntesis como reconocimiento el habla en sus páginas web.

Elaborada por Glen Shires y Hans Wennborg, entre sus casos de uso destacan la búsqueda en la web por voz, la detección de la actividad vocal, los sistemas de diálogo, los videojuegos y la traducción de idiomas.

El estándar proporciona una serie de interfaces que permiten hacer uso de esta funcionalidad de manera sencilla. La más importante de entre todas ellas es la nombrada SpeechRecognition, la cual define métodos para empezar, parar y abortar una interacción con voz por parte del usuario, así como atributos que indican el estado del reconocedor.

Esta API es independiente de la implementación del motor de reconocimiento de habla, pudiendo soportar tanto sistemas en la nube como sistemas locales. Los resultados de reconocimiento se entregan a la página web como un listado de hipótesis, en el cual se refleja la información relevante de cada una de ellas.

En el momento de redacción de este proyecto, la especificación no forma parte de los estándares propuestos por la W3C (World Wide Web Consortium), ni está en el W3C

Standards Track, pero ya se encuentra implementada en sistemas de alta calidad como el proporcionado por Google.

2.3 Clasificación y clustering

El Propósito de este trabajo se puede formular como un problema de clasificación cuyas unidades son los vectores de características (ver sección 2.2.1) de la señal de voz. En esta sección se realizará una exposición de las técnicas del estado del arte relativas a este dominio de la Inteligencia Artificial, con el fin de contextualizar la solución planteada en el proyecto.

El concepto de clasificación se define como una disposición en clases (teórica o de hecho), de objetos, que modela conocimiento capturando su estructura subyacente y relacionando fenómenos entre sí (Mirkin, 2012).

Esta definición se sostiene sobre el axioma de que para hacer generalizaciones, teorías e incluso leyes se puede considerar como grupo un número determinado de objetos similares (Gordon, 1999).

Se podría definir un clasificador como una función que obtiene la clase de una instancia sin clasificar. Todos los clasificadores tienen una estructura de datos almacenada que deben interpretar a la hora de generar la clase para la instancia sin clasificar (Forcada, 2003).

La tarea de cualquier tipo de algoritmo de clasificación es generar un criterio con las siguientes características (Forcada, 2003):

- Que sea preciso. Este requisito es normalmente la característica más importante. La precisión de un clasificador es la probabilidad de clasificar correctamente una instancia seleccionada al azar.
- Que sea comprensible. Dados dos clasificadores con aproximadamente la misma exactitud, se preferirá el más comprensible. Para algunos dominios, como los dominios médicos, la comprensibilidad es crucial. Para otros dominios, como el

reconocimiento de caracteres ópticos, este aspecto no es muy importante.

- Que sea compacto. Aunque está relacionada con la comprensibilidad, una característica no implica la otra.

Los algoritmos que pertenecen a esta área se puede dividir en tres tipos: supervisados, semisupervisados y no supervisados (Ayyasamy, 2013).

2.3.1 Algoritmos supervisados

Estos algoritmos necesitan un conjunto de ejemplos previamente etiquetados con la clase a la que pertenecen, de manera que sea posible inferir un conjunto de reglas que, posteriormente, permitan clasificar nuevas instancias (Ayyasamy, 2013) (Kotsiantis, Zaharakis, & Pintelas, 2007).

La Ilustración 9 muestra el funcionamiento de este tipo de algoritmos a través de un ejemplo en el que se desean clasificar datos web (Ayyasamy, 2013):

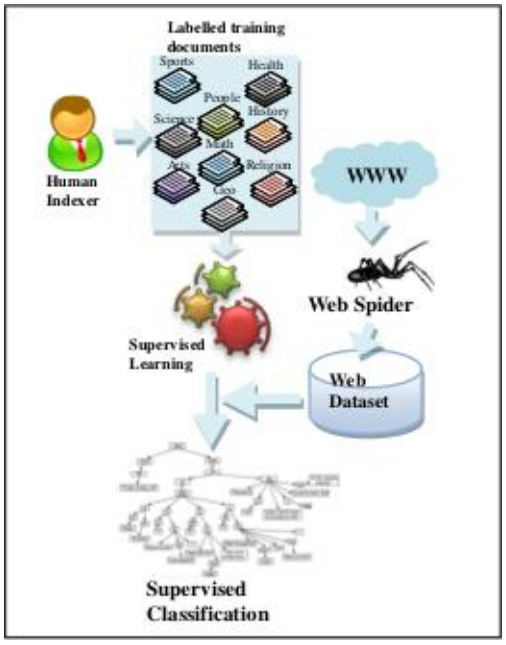

Ilustración 9: Ejemplo de algoritmo supervisado

Entre los principales algoritmos de clasificación supervisada se encuentran (Ayyasamy, 2013) (Forcada, 2003):

- Los árboles de clasificación, cuyo resultado es un conjunto de reglas if-then no independientes entre sí.
- kNN (K nearest neighbour), el cual calcula las distancias entre la muestra a clasificar y cada una de las instancias de los datos de entrenamiento.
- Redes neuronales supervisadas, que constituyen un modelo computacional con un conjunto de propiedades específicas, como son la habilidad de adaptarse o aprender, generalizar u organizar la información, todo ello basado en un procesamiento eminentemente paralelo (Haykin, 1999).
- Naive Bayes, basado en la aplicación del teorema de Bayes, pero con restricciones y suposiciones de partida.
- Redes Bayesianas, que consideran la existencia de una variable especial, la variable a clasificar, que tiene que ser predicha por el resto de las variables.

2.3.2 Algoritmos semisupervisados

Los algoritmos semisupervisados nacen con el objetivo de reducir la carga de trabajo que las personas tienen con los algoritmos supervisados. Para ello, junto a los datos etiquetados se utilizan datos no etiquetados en el proceso de entrenamiento, clasificando a éstos a través de aquéllos y considerando su clase inferida como si fuera hubiera sido etiquetada previamente (Zubiaga, Fresno, & Martínez, 2009).

En la Ilustración 10 se ilustra el funcionamiento de este tipo de algoritmos, con el mismo ejemplo que en la Ilustración 9.

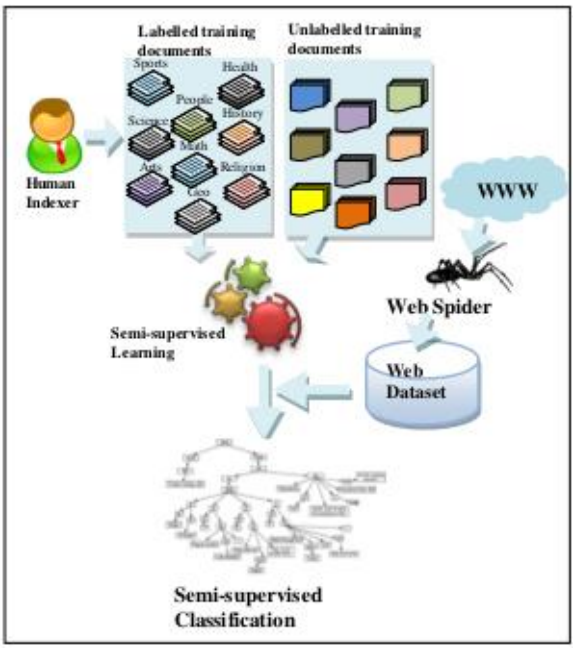

Ilustración 10: Ejemplo de algoritmo semisupervisado

La razón del uso de los documentos no etiquetados y del por qué pueden ayudar al proceso de aprendizaje es expuesta en (Nigam, McCallum, Thrun, & Mitchell, 1998), mediante un experimento con WebKb, una base de datos compuesta por algunas páginas web correspondientes a cursos académicos de universidades, junto a un gran conjunto de páginas web sin etiquetar. Utilizando la información del etiquetado de las primeras, es posible ayudarse de sus características para clasificar aquellas del segundo conjunto que tengan en común una o varias de ellas (Aparicio & Acuña, 2009).

Existen algoritmos como SVM, Naive Bayes y clustering que poseen sus variantes semisupervisadas (Zubiaga, Fresno, & Martínez, 2009) (Ayyasamy, 2013), junto a algoritmos por naturaleza semisupervisados como PLSA (Probabilistic Latent Semantic Analysis).

Las SVM semisupervisadas se conocen también por sus iniciales S3VM, y aunque mejoran a las SVM en problemas binarios, en caso de que sea multiclase el rendimiento no mejora al de un algoritmo supervisado (Zubiaga, Fresno, & Martínez, 2009).

2.3.3 Algoritmos no supervisados

Por último, los algoritmos no supervisados surgieron alrededor del año 1998, y se caracterizan por la ausencia de intervención humana en el proceso de aprendizaje. Por ello, no necesitan conocimiento a priori de cada conjunto de datos a clasificar. En la actualidad esta disciplina es una de las más importantes, ya que no es común disponer de conjuntos de entrenamiento convenientemente etiquetados (Ayyasamy, 2013). Su funcionamiento se ilustra a través de la Ilustración 11.

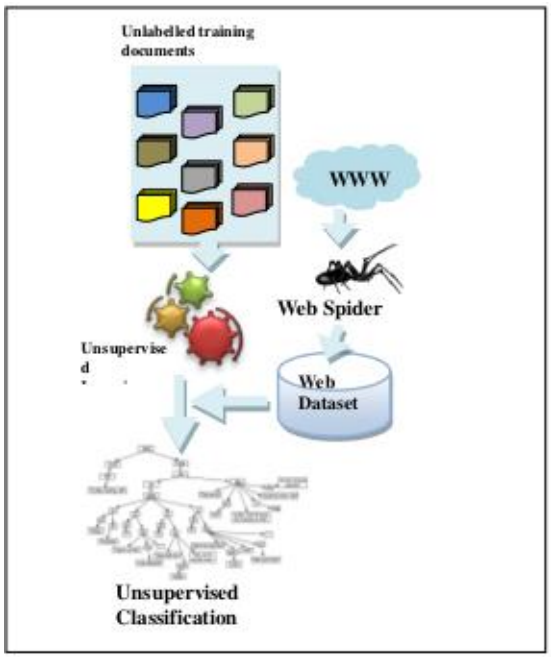

Ilustración 11: Ejemplo de algoritmo no supervisado

En general, la tarea de este tipo de algoritmos es más abstracta que la de los dos anteriores. Para entender la aproximación que realizan, a continuación se expondrá el funcionamiento general de los dos ejemplos de clasificadores no supervisados que más frecuentemente se encuentran en la literatura: ANN (Artificial Neural Networks) en su versión no supervisada (SOM), y clustering (Ayyasamy, 2013).

Las ANN y SOM

Las redes de neuronas artificiales (ANN) tratan de imitar el comportamiento biológico del cerebro humano. Una de las características más llamativas de este último es su capacidad de aprendizaje (Ferrán, 2004). Entre los algoritmos no supervisados basados en redes de neuronas están los mapas auto-organizados o SOM (Self-Organized Maps) (Fustes, 2014).

La Ilustración 12 constituye un ejemplo de la estructura de una red neuronal (Isasi & Galván, Redes de neuronas artificiales: un enfoque práctico, 2004):

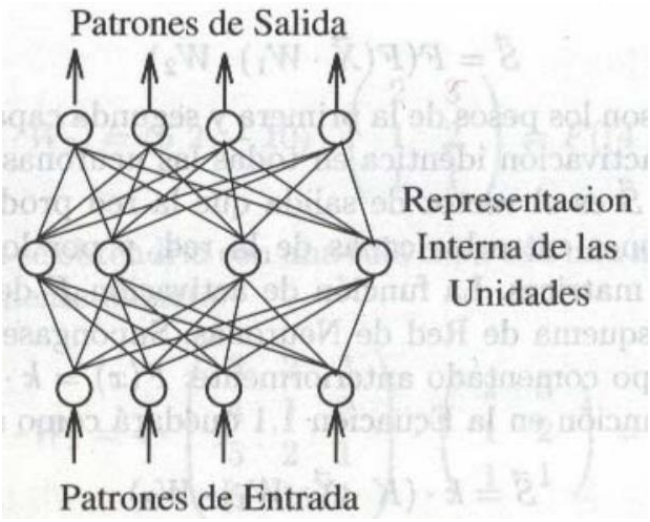

Ilustración 12: Esquema de una red de tres capas totalmente interconectadas

Se denomina *arquitectura de red* a la forma en que las neuronas se conectan entre sí. El primer nivel lo constituyen las neuronas de entrada, que reciben los valores de unos patrones representados como vectores. A continuación hay una serie de capas intermedias, llamadas ocultas, cuyas unidades responden a rasgos particulares que pueden aparecer en los patrones de salida. Finalmente, el último nivel es el de salida, a través de cuyas neuronas se obtiene el resultado final. Las conexiones entre neuronas tienen asociadas pesos, que modifican los valores numéricos de la entrada correspondientes y por tanto alteran el comportamiento de la red. El aprendizaje de una red de neuronas consiste en ajustar estos pesos hasta que los resultados sean significativos (Isasi & Galván, Redes de neuronas artificiales: un enfoque práctico, 2004).

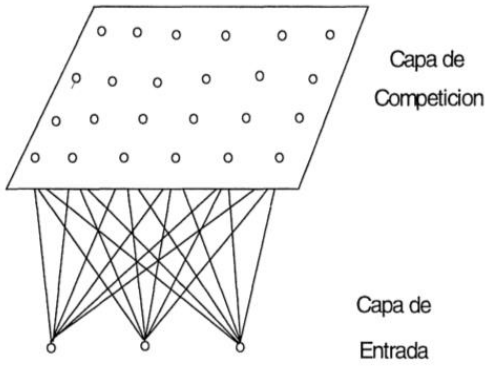

Ilustración 13: Arquitectura del Mapa de Kohonen

Las redes SOM, por su parte, poseen dos capas: una primera de entrada y una segunda de salida (denominada "capa de competición"), en el cual cada neurona de la primera está conectada con cada una de las células de la segunda (Ilustración 13) (Isasi & Galván, Redes de neuronas artificiales: un enfoque práctico, 2004). Las conexiones entre la capa de entrada y la de competición serán una matriz con una ordenación característica, formando una red cuyo objetivo será minimizar la diferencia entre el estímulo de la primera capa y la salida de la red. Para que el algoritmo haga su primera iteración, es necesario inicializar los pesos de la red de manera aleatoria de forma que se vayan modificando conforme el algoritmo aprende (Kohonen, 1990).

Los SOM tienen la capacidad tanto de agrupar datos como de reducir su dimensionalidad, por lo que el número de aplicaciones que tiene es muy alto (Fustes, 2014).

Las redes de neuronas constituyen un método práctico y general para aprender funciones objetivo a partir de ejemplos: son muy robustas frente a los errores en los datos, y para cierto tipo de problemas que implican el aprendizaje a partir de datos del complejo mundo real, como la interpretación de imágenes, el reconocimiento de voz, etc. se encuentran entre los mejores métodos de aprendizaje conocidos (Ferrán, 2004).

A pesar de sus ventajas, las ANN han recibido críticas debido la complejidad de los modelos que generan, lo cual dificulta la comprensión de sus soluciones creando un efecto

de caja negra. Las soluciones obtenidas no poseen una medida de incertidumbre sobre las mismas, por lo que el investigador no puede evaluar de forma simple hasta qué punto los resultados obtenidos por la ANN son aceptables o válidos (Fustes, 2014).

Clustering

El análisis de clusters es la organización de un conjunto de patrones (normalmente representados por un vector de medidas, o un punto en un espacio multidimensional) en agrupaciones, en base a su similitud (Jain, Murty, & Flynn, Data clustering: a review, 1999).

En muchos problemas de análisis de datos, no se dispone de información relevante a priori (por ejemplo, modelos estadísticos), y el resultado final debe ser obtenido haciendo el mínimo número de suposiciones posible. Es en estas circunstancias en las que el clustering es especialmente útil (Jain, Murty, & Flynn, Data clustering: a review, 1999).

Para resolver una tarea a través de este tipo de técnicas, es necesario primero elegir una representación del patrón a clasificar (que puede suponer realizar una extracción y/o selección de características), definir la medida de la distancia entre dos patrones, realizar el clustering y valorar (de ser necesario) los resultados (Jain, Murty, & Flynn, Data clustering: a review, 1999). Estas etapas se ilustran en la Ilustración 14.

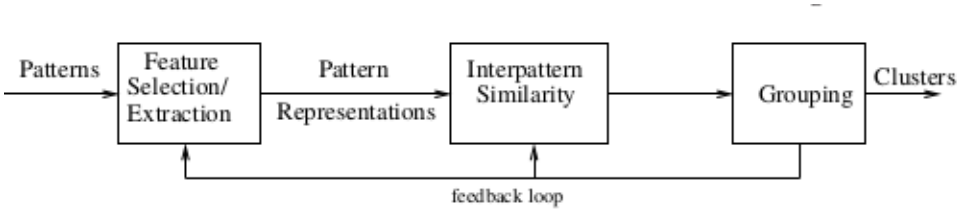

Ilustración 14: Etapas en algoritmos de clustering

Existen dos tipos de algoritmos para realizar el agrupamiento (Grouping, en la Ilustración 14): jerárquicos y particionales (Jain, Murty, & Flynn, Data clustering: a review, 1999).

Jerárquicos

Los algoritmos jerárquicos producen una secuencia de particiones anidadas de los datos de dos maneras distingas: divisiva o aglomerativa (Vidal, 2014).

Los métodos divisivos, también llamados top-down, comienzan considerando a todo el conjunto de datos como un único cluster y en cada nivel dividen cada uno en dos hasta terminar con cada objeto en un único cluster, a menos que se indique algún criterio para dejar de dividir los datos. Los algoritmos divisivos pueden aplicar recursivamente cualquier método particional como k-means, que se verá en detalle más adelante, con k = 2, en cada iteración (Vidal, 2014) (Hastie, Tibshirani, Friedman, & Franklin, 2005).

Por otra parte, la estrategia aglomerativa (también llamada bottom-up) comienza considerando a cada dato como un cluster y en cada iteración mezcla un conjunto de clusters pequeño en un cluster más grande hasta que todos los datos sean considerados un único cluster (Vidal, 2014).

Por lo general, una jerarquía de clusters se representa mediante un árbol denominado dendrograma. El dendrograma (Ilustración 15) muestra cómo los clusters están relacionados y cada nivel de la jerarquía representa una partición particular de los datos.

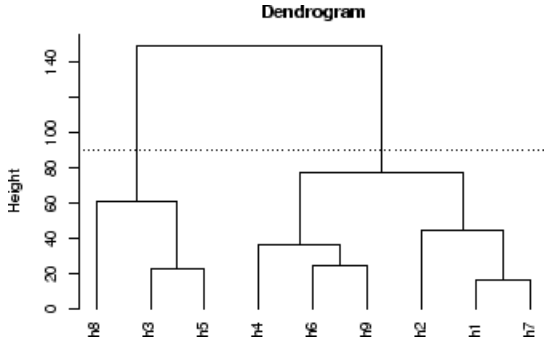

Ilustración 15: Un dendrograma

Ejemplos de este tipo de algoritmos son kNN (citado en Algoritmos supervisados), así como IST (Intra-Cluster Similarity Technique) y CST (Centroid Similarity Technique) (Steinbach, Karypis, Kumar, et al., 2000).

Particionales

Los métodos particionales identifican particiones de los objetos en grupos o clusters de tal forma que todos ellos pertenezcan a alguno de los k clusters posibles, siendo disjuntos, y optimizando un criterio de clusterización (normalmente local, relativo a un subconjunto de patrones) (Larrañaga, Inza, & Abdelmalik, 2008) (Jain, Murty, & Flynn, Data clustering: a review, 1999).

Son más eficientes que los jerárquicos a la hora de procesar grandes cantidades de datos (Jain, Murty, & Flynn, Data clustering: a review, 1999), pero presentan el problema de que es necesario estimar el número de clusters (Larrañaga, Inza, & Abdelmalik, 2008).

Con el fin de ilustrar el funcionamiento de este tipo de métodos, se destallará el algoritmo más popular y a la vez el más simple (Jain A. K., Data clustering: 50 years beyond K-means, 2010): K-means (Steinhaus, 1956) (Lloyd, 1982) (Ball & Hall, 1965) (MacQueen et al., 1967).

K-means

Sea $X = \{x_1, \dots, x_n\}$ un conjunto de *n* puntos *d*-dimensionales para ser agrupados en *k* clusters, $C = \{C_i | i = 1, \dots, k\}$. El algoritmo k-means busca una partición tal que se minimice el error cuadrático entre la media de cada cluster y sus respectivos puntos. Para ello, se formula la siguiente función objetivo a minimizar:

$$J(C) = \sum_{i=0}^{k} \sum_{x \in C_i} \| x - m_i \|^2$$

donde

$$m_i = \frac{1}{nC_i} \sum_{x \in C_i} x_i$$

es el centro del cluster C_i. Minimizar J(C) es un problema NP-hard (incluso para k=2)

(Drineas, Frieze, Kannan, Vempala, & Vinay, 2004), por lo que el algoritmo solamente puede converger en mínimos locales. Existiría la posibilidad de que convergiera en un mínimo global siempre que los clusters estuvieran lo suficientemente separados (Meilua, 2006).

K-means comienza con una partición inicial de *K* clusters y en cada iteración del proceso, se clasifica un nuevo punto y se recalculan los demás centros de los clusters para tener en cuenta esta nueva adición.

El tiempo de complejidad de k-means $O(Nkdm)$, donde *k* es la cantidad de clusters, *N* el tamaño del conjunto de datos, *d* su dimensionalidad y *m* la cantidad de iteraciones. Sus principales desventajas son que es sensible al ruido y a los outliers (los puntos que quedan muy lejos del cluster más cercano) (Vidal, 2014).

Estimación del número de clusters

Existen métodos que permiten estimar el número de clusters de un conjunto de datos. Entre ellos, destacan X-means y Subtractive Clustering.

X-means (Pelleg, Moore, et al., 2000), aplica técnicas de clustering jerárquico sobre K-means con el fin de mejorar sus parámetros iniciales. El primer paso es ejecutar K-means hasta converger, para después determinar si es necesario crear nuevos clusters. Para ello, se divide cada cluster en dos y se alejan entre sí una distancia proporcional al tamaño de la región en la dirección de un vector elegido al azar y en sentidos opuestos. En la región definida por los dos nuevos clusters se ejecuta K-means (con $k = 2$), y se determina si el cluster padre modelaría la distribución de puntos de igual manera que lo hacen los dos clusters hijos por separado, en cuyo caso se eliminarían los dos hijos y permanecería el padre.

Para determinar cuál de las dos opciones (dos clusters hijos o cluster padre) modela mejor la distribución de puntos correspondiente, se utiliza el modelo BIC.

Este modelo calcula la probabilidad de que el modelo M_j, que en este caso corresponde a una solución del algoritmo K-means con un *k* determinado, ocurra junto al conjunto de datos clasificados por K-means, utilizando el criterio de Schwarz (Kass & Wasserman, 1995).

Subtractive Clustering, por su parte, es descrito en (Chiu, 1994) y se basa en el algoritmo Mountain Method (Yager & Filev, 1994) para crear una red de puntos que, en función de la densidad de la región que ocupen, se convertirán en centros de cluster o no. Computacionalmente mejora Mountain Method, que toma en cuenta puntos de la red sin datos asociados, y también evita la necesidad de definir una resolución para ella, ya que el número de puntos de la red a evaluar es simplemente el número de vectores de datos.

3 Planteamiento

Como se ha dicho en la sección 1, el proceso de investigación del cual este trabajo forma parte pretende localizar las agrupaciones de vectores de características y establecer si existe una relación unívoca entre ellos y los segmentos de habla a los que corresponden. Si así fuera, sería posible sustituir las descripciones que realizan los GMM por las que realizan estas burbujas.

Los modelos GMM son muy sensibles a las distorsiones que se producen en el espectro de la señal de audio (lo cual dificulta la robustez de los reconocedores que los emplean), por lo que de ser posible llevar a cabo esta sustitución, cabría esperar una mejora considerable de los resultados de reconocimiento actuales. Los modelos HMM emplearían unidades deterministas que reducirían la dimensionalidad del espacio de búsqueda, por lo que los resultados serían obtenidos más rápidamente y además de manera mucho más precisa.

3.1 Definición del problema

Este proyecto aborda la primera fase de esa investigación, que consiste en establecer si existe la posibilidad de identificar regiones de alta densidad (denominadas burbujas en este proyecto) en el espacio de vectores de características. En este caso, se utilizarán coeficientes cepstrales (MFCC) debido a que son las más populares en el estado del arte del reconocimiento de habla.

Teniendo en cuenta que estos coeficientes son estadísticamente independientes entre sí, y asumiendo que las burbujas siguen una distribución gaussiana, estas últimas se podrán definir a través de su centroide (las medias) y de sus semiejes (representados por la matriz de covarianzas).

Su búsqueda habrá de realizarse a través de algoritmos no supervisados, debido a que la investigación llevada a cabo sobre el estado del arte en relación a este problema no ha sido capaz de proveer de más información sobre las estructuras de las burbujas.

La expectativa de que estas regiones no varíen en función de elementos aleatorios se funda sobre el análisis realizado en el estado del arte, en el cual se describe un proceso natural sin elementos de aleatoriedad en él que refleja la realidad de la fonación y la percepción del sonido humanas. Sin embargo, los resultados obtenidos por el algoritmo que resuelva el problema deberían reflejar la forma en que se realiza cada alófono presente en los datos procesados, lo cual será de importancia para las siguientes fases de la investigación que aborden el problema de la generalización de estas características entre personas diferentes.

Ya que los bifonemas contienen una coarticulación entre dos alófonos, constituirán la unidad de análisis de este proyecto.

3.2 Propuesta de solución

El problema se abordará a través de un algoritmo no supervisado que entregue resultados deterministas. Las alternativas planteadas en el estado del arte son las redes de neuronas y los algoritmos de clustering.

Las redes de neuronas tienen el problema de que sus resultados no modelan una distribución normal, por lo que no son útiles dado que las burbujas se asumen como normales. Por otra parte, existen algoritmos de clustering que sí cumplen con esta propiedad y también son deterministas, por lo que será uno de ellos el que se utilice para resolver la cuestión.

Entre ellos existen dos categorías que procesan la información de maneras distintas. Mientras que los algoritmos pertenecientes a la categoría de jerárquicos producen particiones anidadas unas dentro de otras, los particionales tratan de optimizar un criterio de clusterización en un subconjunto de los patrones presentados a la entrada de manera que se obtengan particiones disjuntas entre sí.

En el problema que atañe a este proyecto, los patrones que se quieren clasificar están repartidos por el espacio de características. Dado que el objetivo es identificar regiones de

alta densidad en él, existe un claro criterio de clusterización que provoca que la elección de un algoritmo particional.

El algoritmo particional más popular es K-means, el cual requiere de un parámetro inicial que indique el número de clusters presentes en los datos. Debido a que para este problema ese número se desconoce, será necesario emplear otro algoritmo que lo estime por sí mismo.

En el estado del arte se presentaron dos algoritmos de este tipo, X-means y Subtractive Clustering. El objetivo de este último es exactamente el propuesto para el planteamiento del problema (identificar regiones de alta densidad), por lo que es el algoritmo elegido para llevar a cabo la experimentación y las pruebas.

3.3 Subtractive clustering

Subtractive Clustering (Chiu, 1994) tiene como objetivo encontrar tantas agrupaciones como haya en los datos a clasificar. Su característica más relevante con respecto a otros algoritmos de clasificación no supervisada es que no necesita conocer de antemano el número de clusters que hay en ellos, puesto que lo estima empleando criterios locales (el número de puntos concentrados en una región determinada). Para su correcto funcionamiento, será necesario normalizar los datos y proponer un “radio de vecinos” propio de cada problema concreto.

El primer paso del algoritmo es calcular el potencial de cada punto del conjunto de datos (en el caso de este proyecto, de cada vector de características), el cual medirá la densidad de la región en la que se encuentra a través de la consideración de las distancias de cada punto a todos los demás.

De esta manera, se elegirá como primer centro de cluster aquel cuyo potencial sea el más alto. Obteniendo una muestra representativa de los puntos del cluster mediante la selección de aquellos puntos cuya distancia del centro sea menor que el radio de vecinos, podrá caracterizarse el cluster mediante la matriz de covarianzas de esos puntos y el

centroide como media.

Con el primer cluster identificado, se eliminan los puntos de la muestra representativa, se realizan modificaciones de ajuste a los potenciales del resto de puntos (para no considerar los puntos eliminados) y se vuelve a repetir el proceso eligiendo como siguiente centro aquel punto cuyo potencial sea el más alto.

La Ilustración 16 muestra el diagrama de flujo del algoritmo a alto nivel.

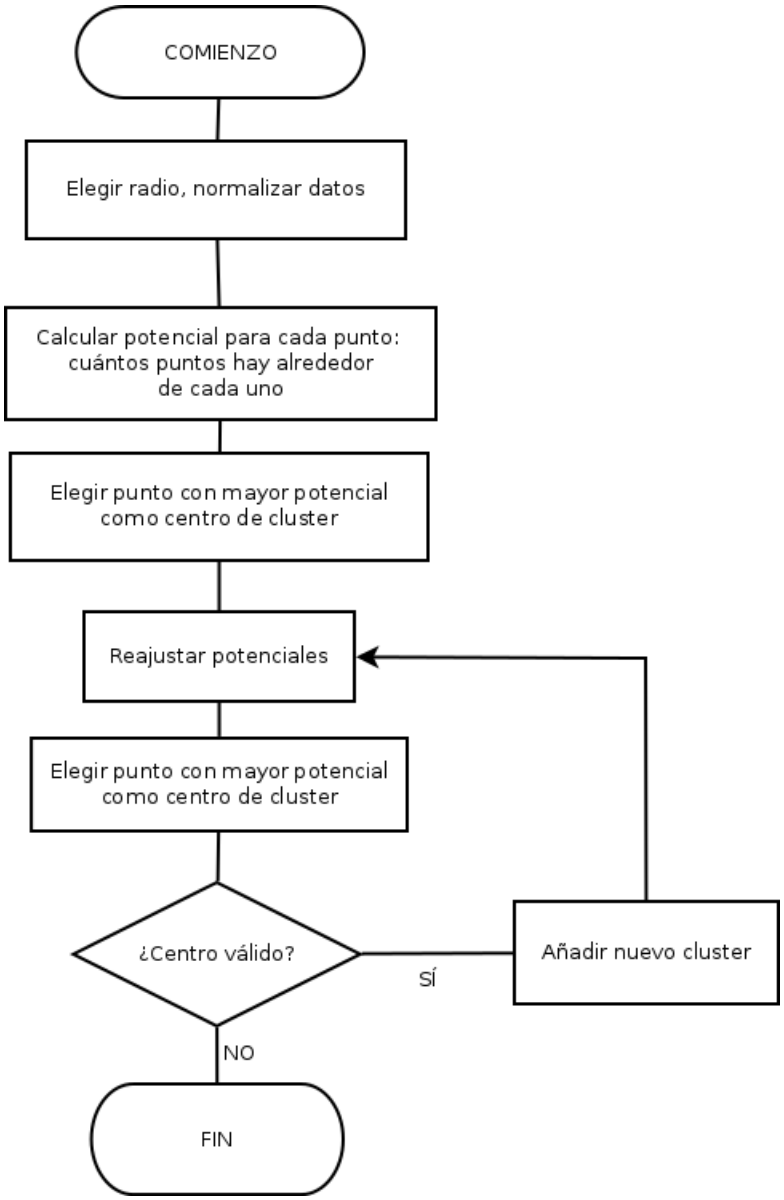

Ilustración 16: Diagrama de flujo de Subtractive Clustering

El algoritmo finaliza cuando el máximo potencial es menor que un cierto umbral definido en función del potencial del primer centro, ya que eso significa que no hay datos que tengan un potencial mayor que éste y por tanto, todos los demás se verán también rechazados.

Para ver los detalles de este proceso, primero se introducirán los algoritmos sobre los cuales Subtractive Clustering se basa, para después especificar el funcionamiento de cada etapa a través de sus ecuaciones matemáticas. Finalmente, se expondrá una modificación llevada a cabo sobre el algoritmo para resolver el problema de la estimación del radio inicial.

3.3.1 Algoritmos base

Para comprender este proceso, es necesario formular la relación que existe entre este algoritmo con FCM (fuzzy c-means) y Mountain Method.

FCM es un algoritmo iterativo de optimización que minimiza la función de coste

$$J = \sum_{k=1}^{n} \sum_{i=1}^{c} \mu_{ik}^{m} \parallel x_k - v_i \parallel^2$$

donde *n* es el número de puntos presentes en el conjunto de datos, *c* el número de clusters, x_{k} el punto *k*-ésimo, v_{i} el centro del cluster *i*-ésimo, μ_{ik} el grado de pertenencia del punto *k*-ésimo al cluster *i*-ésimo, y *m* una constante mayor que 1 (típicamente, *m* = 2).

El grado de pertenencia μ_{ik}está definido por

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{r} \frac{\parallel x_k - v_i \parallel}{\parallel x_k - v_j \parallel}^{\frac{2}{m-1}}}$$

El número de clusters a obtener y la aproximación inicial de sus centros determinarán en gran medida la calidad de las soluciones obtenidas a través de este algoritmo. Debido a la dificultad para estimarlos manualmente, se hace necesario proponer algoritmos capaces de estimar el número de clusters presentes en un conjunto de datos.

Mountain Method es uno de estos algoritmos. Empieza componiendo una red de puntos dentro de la región definida el parámetro de entrada, y computa un potencial para cada uno de ellos basándose en sus distancias con respecto a los demás, de manera que tendrán potencial alto si hay muchos puntos alrededor.

La idea clave del método es que una vez el primer centro de cluster es encontrado, los potenciales de los demás puntos de la red se ven reducidos de acuerdo a su distancia con respecto al centro del cluster. Con estos valores se procede a elegir el siguiente cluster, que de nuevo será aquel que tenga el mayor potencial, y así sucesivamente hasta que el máximo potencial esté por debajo de un cierto valor.

Aunque este método es simple y efectivo, su complejidad computacional es exponencial con respecto a las dimensiones del problema. Subtractive Clustering propone una solución a este problema: incluye en la red de puntos únicamente aquellos que correspondan a datos. De esta manera también se elimina la necesidad de especificar la resolución de la red de puntos, donde se han de balancear la precisión del algoritmo y la complejidad computacional: con la reducción de la red, la complejidad crece linealmente con respecto a las dimensiones del problema.

3.3.2 Estimación de clusters

Dada una colección de *n* puntos de datos normalizados $\{x_1, x_2, \dots, x_n\}$

en un espacio *M*-dimensional de manera que los puntos estén limitados por el hipercubo correspondiente, se puede considerar un posible centro de cluster a cada punto de datos. Definiéndose la medida del potencial de cada punto de datos x_i como

$$P_i = \sum_{j=1}^{n} e^{\alpha \|x_i - x_j\|^2}$$

donde

$$\propto = \frac{4}{{r_a}^2}$$

y r_a es una constante positiva, cada potencial se verá determinado por la distancia al resto de puntos de datos: aquel que tenga muchos a su alrededor tendrá un potencial alto. La constante r_a determinará aquellos puntos sobre los cuales la influencia del centro encontrado es mayor inicialmente.

Después de calcular los potenciales de todos los puntos de datos, se selecciona aquel que tenga mayor potencial como el primer centro de cluster. En función de su ubicación $x_1^\star$ y su potencial $P_1^\star$, se puede redefinir el cálculo de los demás potenciales como

$$P_i \leftarrow P_i - P_1^\star e^{-\beta \|x_i - x_1^\star\|^2}$$

donde

$$\beta = \frac{4}{{r_b}^2}$$

y r_b es una constante positiva. Esto significa que la cantidad de potencial que se le resta con cada centro de cluster encontrado es proporcional a la distancia con respecto a éste. La constante r_b es el radio que define los puntos de un cluster, es decir, el radio de vecinos, y ha de equilibrarse con r_a para que los clusters no estén demasiado cercanos en el espacio: por ello se propone establecer una relación tal que $r_b = 1.5 r_a$.

En cada iteración, se decide si el punto de mayor potencial disponible se añade o no a la lista de clusters. En caso de que su potencial sea mayor que una cierta proporción $\bar{\varepsilon}$del máximo potencial inicial, se añade. Si por el contrario fuera menor que otra proporción diferente $\underline{\varepsilon}$ del máximo potencial inicial, se rechazaría y el algoritmo terminaría. Si no ocurriera ninguno de los dos casos, se mediría su distancia con respecto al resto de clusters ya añadidos y se añadiría siempre que la proporción entre el radio de vecinos y la distancia mínima con respecto a los clusters sea mayor que 1 menos la proporción entre el potencial del punto y el máximo potencial inicial. Los valores de las proporciones suelen ser $\bar{\varepsilon}$= 0.5 y $\underline{\varepsilon}$= 0.15.

3.3.3 Distancia de Mahalanobis

El algoritmo tal y como ha sido expuesto tiene el problema de que es necesario estimar el valor del radio para cada problema concreto.

Con el objetivo de reducir la importancia de este paso, se realizó una modificación sobre el flujo del algoritmo que está reflejado en la Ilustración 17:

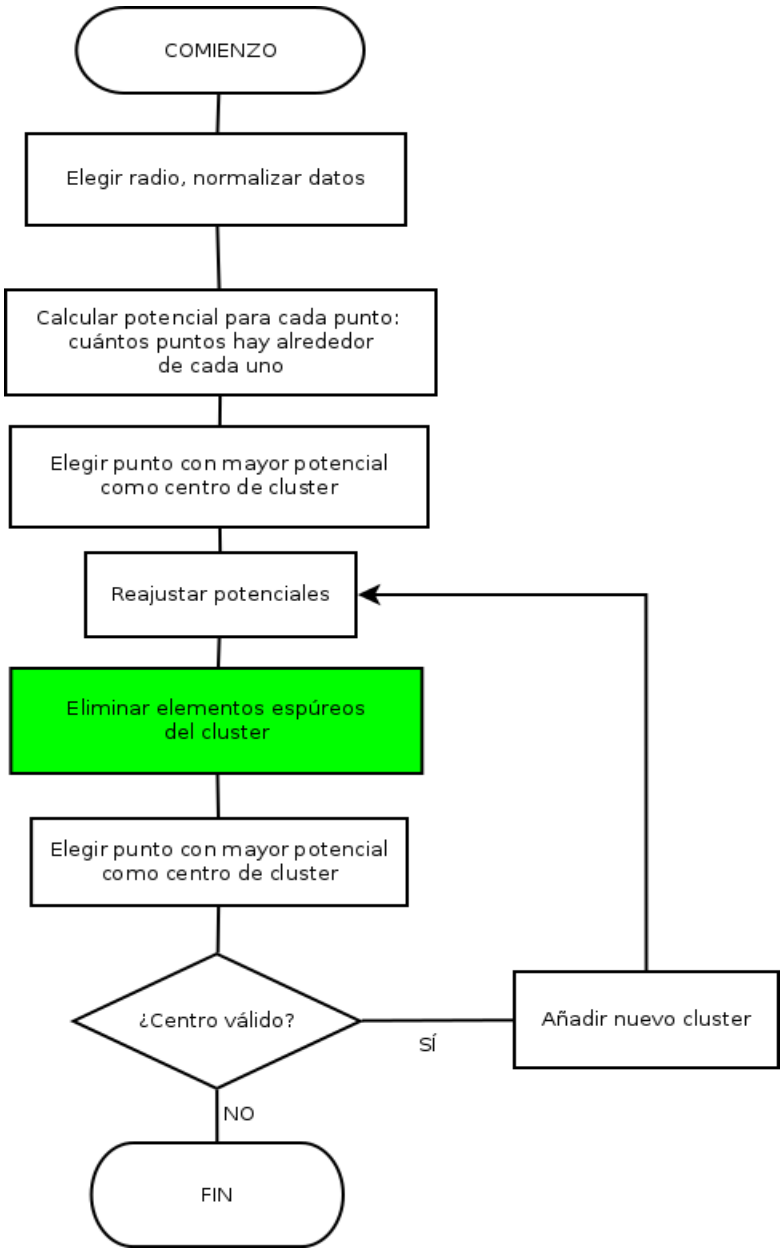

Ilustración 17: Subtractive Clustering con eliminación de elementos espúreos

Como se puede observar en ella (Ilustración 17) se introduce un nuevo paso que elimina elementos que pueden ser considerados espúreos dentro del cluster identificado. Para ello se hace uso de una medida de distancia que tiene en cuenta cada una de las dimensiones del espacio, así como la correlación entre los puntos que lo componen: la distancia de Mahalanobis (Xiang, Nie, & Zhang, 2008).

Una vez se han obtenido los puntos que pertenecen al nuevo cluster, se iterará hasta que todos los puntos del nuevo cluster estén dentro de una cierta distancia de Mahalanobis con respecto a la distribución normal definida por el centro del cluster y su matriz de covarianzas. Esta distancia se calcula a través del análogo de la regla de las tres sigmas para espacios multidimensionales, la cual depende únicamente del número de dimensiones del espacio.

El objetivo final es que cada cluster esté compuesto únicamente por aquellos puntos que se encuentren en el rango del 95% de la normal definida por el cluster. Así, en cada iteración de este paso se irá reduciendo la burbuja hasta que se cumpla este criterio.

En caso de que no se cumpla, el algoritmo no convergerá y por tanto no podrá ser aplicable al problema.

3.4 Plan de experimentación y pruebas

Entre los parámetros de Subtractive Clustering, el más importante es el radio de vecinos. Mientras que los demás parámetros tienen valores estándar propuestos en (Chiu, 1994), este radio determinará cuántos clusters se crearán y qué distribución estadística los modelarán, por lo que la experimentación estará centrada en él.

Por este motivo se procedió a realizar diferentes experimentos con diferentes valores del radio, buscando probar que existen clusters comunes entre ellos que también pueden ser utilizados para definir bifonemas en el conjunto de datos a clasificar.

Cada experimento se compone de 3 pasos: primero se ubican los vectores de características en el tiempo, completando la información de las etiquetas generadas

manualmente de los ficheros de audio correspondientes. A continuación se ejecuta el algoritmo con la matriz de vectores de coeficientes cepstrales, con un radio concreto, obteniendo como resultado el conjunto de centros de clusters y sus ubicaciones en el tiempo con respecto al fichero de audio al que corresponden. Con esa información y el conjunto de etiquetas de bifonemas preparado en la base de datos, se asocian los clusters con los bifonemas.

Finalmente se computa el número de clusters generados para ese radio, cuántos de ellos pertenecen exclusivamente a un bifonema y cuántos bifonemas con menos de cuatro clusters asociados hay. Para ello se elaborará una matriz binaria que modele la relación de pertenencia entre bifonemas y clusters.

Cuantos menos clusters estén asociados a un bifonema, más posible es que se haya encontrado una buena caracterización para él. También sería interesante que el número de clusters propios de cada fonema fuera alto, para que la diferencia entre los bifonemas sea lo más alta posible.

4 Desarrollo experimental

El desarrollo de este proyecto ha transcurrido en diferentes etapas: primero un análisis teórico del problema, durante el cual se estableció el alcance de la experimentación y se documentaron las alternativas y los requisitos del mismo; a continuación se implementaron las soluciones propuestas, se experimentó y se establecieron los valores de los parámetros de la algorítmica; por último, se obtuvieron los resultados y se extrajeron conclusiones de ellos.

Durante la primera etapa del desarrollo de este proyecto se realizó una búsqueda de algoritmos en el estado del arte que pudieran servir para el propósito planteado.

Se exploraron diferentes aproximaciones al problema de la clasificación. Una de ellas utilizaba la programación genética para diseñar un algoritmo que minimizara una suma ponderada entre el número de vectores de características sin clasificar y la separación total entre ellos.

Otro campo de la inteligencia artificial que se sondeó fue el de las redes de neuronas: se buscaba la manera de extraer los datos más relevantes en un solo vector de características, para ser capaces de clasificar en función de ellos. Esta idea se basaba en la sospecha de que cada vector de características escondía ciertos datos que, aislados convenientemente, arrojarían luz acerca de qué determinaba que un segmento de habla perteneciera a un fonema u otro.

Una tercera idea que se contempló fue utilizar fractales. Los patrones de que están compuestos varían sus distancias en cada escala, y de la misma manera se pensó que podrían variar las representaciones en forma de vectores de características de los fonemas: ya que los fonemas varían en su duración, podría existir una estructura subyacente que se repitiera en todos ellos y que los identificara como fonemas.

Estos tres algoritmos fueron planteados durante una reunión de seguimiento del proyecto,

en la cual se acordó limitar su alcance al empleo y leve modificación de algoritmos ya existentes.

Una vez descartadas estas ideas, se procedió a buscar uno ya diseñado e implementado que pudiera clasificar de manera no supervisada y con el menor número de parámetros de entrada posible vectores de características. El tutor de este trabajo propuso uno que él había estado diseñando y que se basaba en técnicas de clustering particional, el cual constituyó el punto de partida para encontrar implementaciones cercanas a él o conceptualmente similares.

Explorando la batería de algoritmos que el software Matlab® implementa, se observó la existencia de uno denominado Subtractive clustering que cumplía con casi todos los requisitos: los parámetros de entrada eran pocos, todos menos uno tenían valores por defecto que se indicaban como razonables para la mayoría de experimentos, y pertenecía a la familia de los algoritmos de clustering particional. La única diferencia radicaba en que el algoritmo propuesto inicialmente realizaba un paso extra a cada iteración donde se utilizaba la distancia de Mahalanobis para eliminar vectores espúreos.

Debido a este inconveniente se continuó buscando algoritmos que se encontraran completamente implementados y listos para ser probados. (Li & Sim, 2014) y (Jain R. , 2012) proponían dos, pero como no existía implementación disponible se descartaron.

Finalmente se eligió el algoritmo presente en Matlab®, y se comenzó a modificarlo para incorporar el paso adicional en cada iteración.

Antes de empezar la evaluación, era necesario disponer de una base de datos para alimentar al algoritmo. Se consideró utilizar una disponible de manera inmediata, como la utilizada en (Puente, 2014). Finalmente fue descartada ya que presentaba un sesgo (debido a su orientación hacia el reconocimiento de locutor) cuya corrección requería un esfuerzo de reetiquetado que se consideró innecesario.

Entre los criterios para elegir una nueva, estaba que el audio fuera lo más limpio posible (para facilitar así el etiquetado), que contuviera locuciones de varios usuarios (para que la

experimentación fuera más robusta), y cuya obtención fuera accesible. Fue por ello por lo que se escogió MICROAES en su versión 1, elaborada por Antonio Rincón Rincón y Elena Hernandez para la empresa ATLAS (Applied Technologies on Language and Speech, S.L.), que además de las características anteriores, tenía el texto ya transcrito.

Antes de empezar a realizar los primeros experimentos, fue necesario etiquetar los datos para que los vectores de características pudieran ubicarse en el tiempo y relacionarse con los segmentos del discurso (unidades de análisis) correspondientes.

Este paso comprendía dos tareas: primero había que decidir qué unidad de análisis se iba a utilizar para el proyecto, y después etiquetar los audios en fragmentos correspondientes.

Las opciones para la primera cuestión eran básicamente dos: bifonemas y trifonemas. Los fonemas estaban descartados al ser imposible marcar su comienzo y su fin, y tanto bifonemas como trifonemas evitaban este problema ya que las etiquetas se situarían en los puntos medios de cada fonema. Finalmente se decidió utilizar bifonemas en detrimento de los trifonemas por dos razones: a igual cantidad de audio etiquetado hay más representación porcentual de bifonemas que de trifonemas, y el proceso de etiquetar bifonemas se hacía más natural que etiquetar trifonemas.

Así se procedió a etiquetar 20 minutos de audio (según el criterio definido por el Apéndice 1 de (Puente, 2014), un límite pequeño para una investigación profunda pero lo suficientemente grande como para que los experimentos de este proyecto identificaran tendencias útiles para posteriores investigaciones.

Durante el proceso de etiquetado se decidió incorporar algunas mejoras a la versión beta del Pitch Marker implementada para (Puente, 2014). Mientras tanto, se identificó la necesidad de adquirir conocimientos más profundos de las técnicas básicas utilizadas en el dominio del procesamiento del habla, así que se revisó documentación acerca de HEQ, HMM y MFCC, así como de distribuciones gaussianas multivariables (Viele, 2008).

Con estos conceptos más claros, se comenzó a estudiar la implementación del algoritmo escogido. Era necesario asegurarse de que el proceso efectivamente implementaba un

algoritmo determinista. Se inspeccionó el algoritmo visualmente, para asegurarse de que no había ningún elemento de aleatoriedad, y se probó con pequeños conjuntos de datos para confirmarlo.

El algoritmo recibiría como datos de entrada vectores de coeficientes cepstrales de los audios, y se experimentaría con el radio de vecinos por ser éste el único que no tenía un valor propuesto en (Chiu, 1994).

Para obtener los vectores se decidió utilizar una configuración habitual en el estado del arte del reconocimiento de habla, formada por 39 coeficientes cepstrales: 12 coeficientes de primer orden, sus 12 velocidades y sus 12 aceleraciones, junto a la energía, su velocidad y su aceleración. Se buscó una implementación en Matlab® (dado que en ese momento era el entorno de programación elegido) hallándose MIRtoolbox en su versión 1.5 (2013), capaz de extraer los 39 coeficientes cepstrales de los ficheros de audio etiquetados de la base de datos.

Con la idea de automatizar al máximo el proceso, se contempló la posibilidad de eliminar la dependencia radio de vecinos que el algoritmo poseía. Para ello, se propusieron dos soluciones: utilizar la distancia de Mahalanobis para que reducir los clusters obtenidos hasta que todos sus puntos estuvieran dentro de un cierto límite, y hacer que el parámetro fuera actualizado dinámicamente en función de los demás.

La primera opción finalmente se descartó ya que en pruebas con pequeñas cantidades de datos se obtenían diferentes resultados para diferentes parametrizaciones, y la experimentación con grandes volúmenes no conseguía que las soluciones convergieran en un tiempo computacional razonable.

La segunda opción, por su parte, exigía modificar el algoritmo inicial de tal manera que unos parámetros influyeran en otros, lo cual cambiaría la naturaleza del algoritmo ya que originalmente constituyen límites y no semillas para nuevos cálculos.

En este punto, ya se disponía de datos etiquetados y del planteamiento de la algorítmica general. Se desarrolló software que sirviera de enlace entre ambos, y se comenzó a

experimentar. Se generaron los primeros modelos, y se procedió a introducir todos los audios etiquetados y producir los primeros resultados totales. Resultó imposible: las dos versiones de Matlab® de que se disponía (la de estudiantes y la profesional de 32 bits) no podían manejar matrices de datos tan grandes como era necesario para procesar todo el audio seleccionado. Se decidió dejar de utilizar Matlab® y buscar un lenguaje que permitiera la transición sencilla y no presentara esta limitación.

Se consideró Java, R y Python. R se descartó por ser un lenguaje demasiado especializado en estadística. Entre Java y Python se escogió este último debido a que es un lenguaje en boga dentro del mundo científico, y posee muchas librerías y tutoriales que facilitan la transición desde Matlab®.

La migración de Matlab a Python fue asumible gracias a que el algoritmo de clusterización estaba también implementado en Python. La orientación a objetos de Python y sus librerías científicas (numpy y scipy) ayudaron a refactorizar el código hecho para Matlab® y crear un proyecto más limpio y fácil de mantener.

Sin embargo, se hizo necesario repetir el proceso de inspección y prueba con la nueva implementación del algoritmo, y realizar las modificaciones oportunas. Se detectaron algunos problemas con el código original en Matlab®, así que gracias a que era software libre pudo ser modificado y adaptado convenientemente. Para comprobar su correcto funcionamiento, se utilizó una batería de tests que compararan los resultados del algoritmo en Matlab® con los obtenidos por Python. Este proceso también necesitó de una etapa de aprendizaje de Python como lenguaje, ya que el proyectando no lo había utilizado nunca.

Como se ha señalado anteriormente, la adaptación del algoritmo para que los resultados no dependieran de ningún parámetro no se descartó definitivamente hasta el final. Con la implementación en Python funcionando, se realizaron los mismos experimentos que sin esta adaptación y no se obtuvieron resultados: el algoritmo no convergía en un tiempo razonable.

Finalmente se ejecutaron los experimentos y se recogieron los datos necesarios para establecer las conclusiones. Para ello se desarrolló código que serializara los clusters, los relacionara con etiquetas y serializara estas etiquetas, de forma que se pudiera analizar un solo fichero para cada experimento.

Cada experimento proporcionaría una matriz binaria relacionando bifonemas y burbujas, a través de la cual se podrían realizar múltiples cálculos que confluyeran en una serie de gráficas y tablas de resultados.

5 Resultados de las pruebas

5.1 Valores de potenciales

Durante la primera fase del algoritmo escogido (descrito en la sección 3.3) para la resolución del problema planteado, se miden los potenciales de cada vector de características.

Los experimentos han mostrado que estos potenciales dependen fuertemente del radio de vecinos que se utilice, pudiéndose generar tanto distribuciones casi uniformes como muy variables. El mero hecho de que con diferentes radios varíe el valor del máximo potencial ya indica que hay regiones del espacio de vectores de características que están más pobladas que otras.

En la Tabla 1 y la Ilustración 18 pueden observarse los valores del máximo y del mínimo potencial para cada uno de los radios. En el gráfico destaca una notable diferencia entre los resultados del experimento con radio 0.3 y los de los demás, aunque el mínimo valor de un potencial es prácticamente idéntico en todos los casos, lo cual indica que hay puntos que están bastante aislados de los demás.

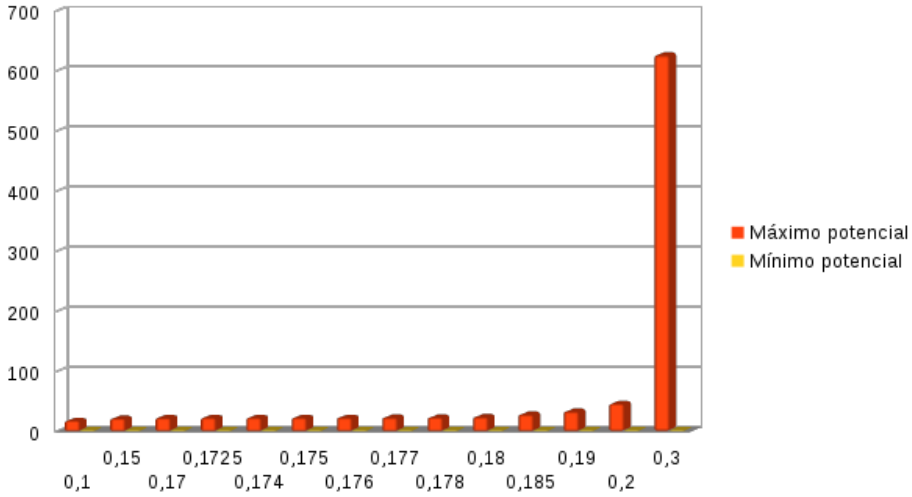

Ilustración 18: Máximo y mínimo potencial

Tabla 1: Valores del máximo y mínimo potencial

Radio	Máximo potencial	Mínimo potencial
0.1	17.0756337	2.0
0.15	20.7909296	2.0
0.17	21.9568786	2.0
0.1725	22.0949671	2.0
0.174	22.1815607	2.0
0.175	22.2388365	2.0
0.176	22.2957556	2.0
0.177	22.3523229	2.0
0.178	22.4085432	2.0
0.18	23.2034928	2.0
0.185	27.4202374	2.0
0.19	32.4877566	2.0
0.2	45.6521376	2.00000004
0.3	626.054379	2.00682039

5.2 Número de burbujas

La Tabla 2 y la Ilustración 19 muestran que la diferencia entre la mayor cantidad de burbujas por bifonema y la media de burbujas por bifonema en cada uno de los radios es lo suficientemente grande como para que, teniendo en cuenta el número de burbujas totales reflejados en la Tabla 3, se pueda constatar el hecho de que los bifonemas tienen un número variable burbujas asociadas, lo cual confirma la existencia de regiones de alta densidad.

Tabla 2: Número de burbujas

Radio	Máximo número de burbujas	Media número de burbujas
0.174	1675.0	34.0606061
0.175	1684.0	34.184573
0.176	1677.0	34.2038567
0.177	1672.0	34.1873278
0.178	1677.0	34.15427
0.18	1589.0	32.0688705
0.185	1072.0	21.9862259
0.19	679.0	14.0688705
0.2	1685.0	34.6942149
0.3	1685.0	34.6969697

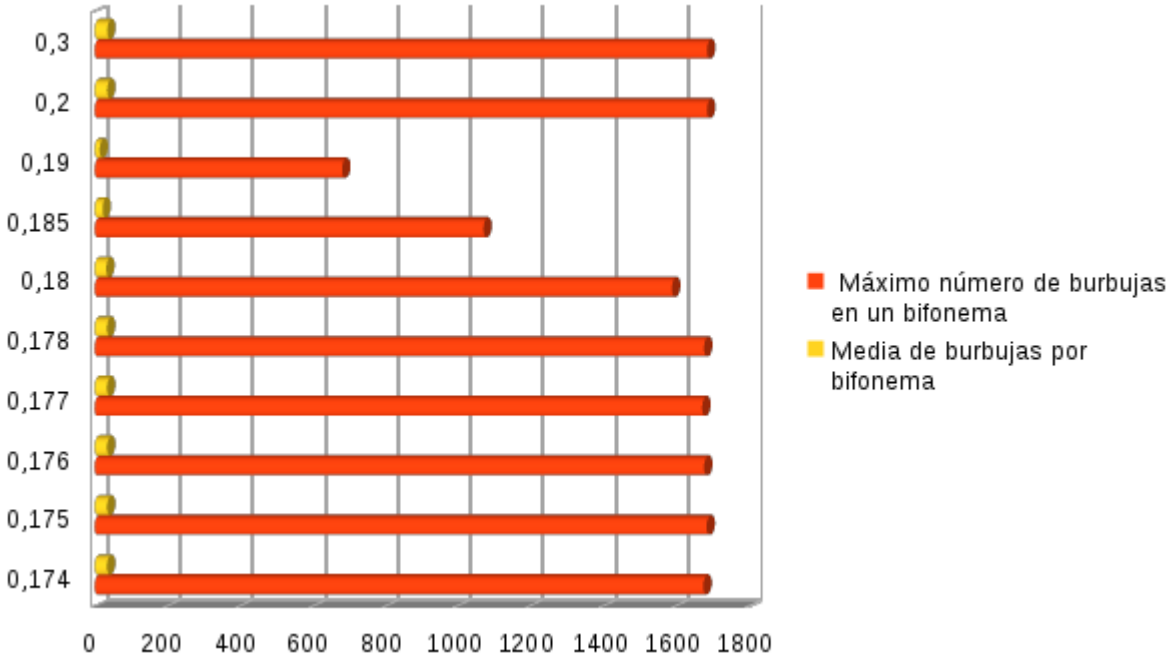

Ilustración 19: Número de clusters en función del radio

5.3 Bifonemas buenos

Sin embargo, en el marco más amplio de la investigación de la cual este proyecto forma parte, existen datos que proporcionan información acerca de la posibilidad de que burbujas de este tipo puedan caracterizar unívocamente bifonemas.

Si se considera un buen bifonema aquel al que están asociadas como máximo cuatro burbujas, y definiendo una burbuja única como aquella que solamente se asocia a un bifonema, la Tabla 3, la Ilustración 20 y la Ilustración 21 muestran resultados sorprendentes:

Tabla 3: Burbujas y bifonemas

Radio	Bifonemas	Bifonemas buenos	Burbujas	Burbujas únicas
0.19	363.0	25.6198347%	5107.0	100.0%
0.185	363.0	21.7630854%	7981.0	100.0%
0.18	363.0	18.1818182%	11641.0	100.0%
0.174	363.0	17.9063361%	12384.0	100.0%
0.2	363.0	17.630854%	12594.0	100.0%
0.3	363.0	17.630854%	12595.0	100.0%
0.175	363.0	17.0798898%	12409.0	100.0%
0.176	363.0	17.0798898%	12416.0	100.0%
0.178	363.0	16.8044077%	12398.0	100.0%
0.177	363.0	16.8044077%	12410.0	100.0%

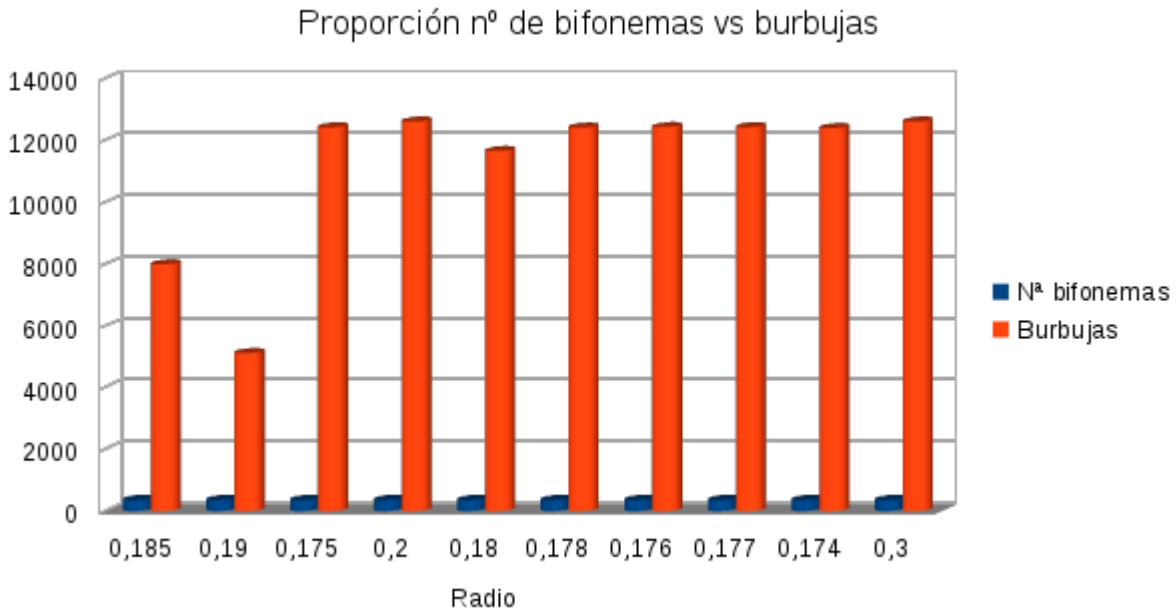

Ilustración 20: Proporción nº de bifonemas vs burbujas

En la Ilustración 20 se puede apreciar el gran volumen de burbujas generadas, comparándolo con el número total de bifonemas a partir de los cuales fueron extraídas: siempre mayor a 10 veces esta cantidad.

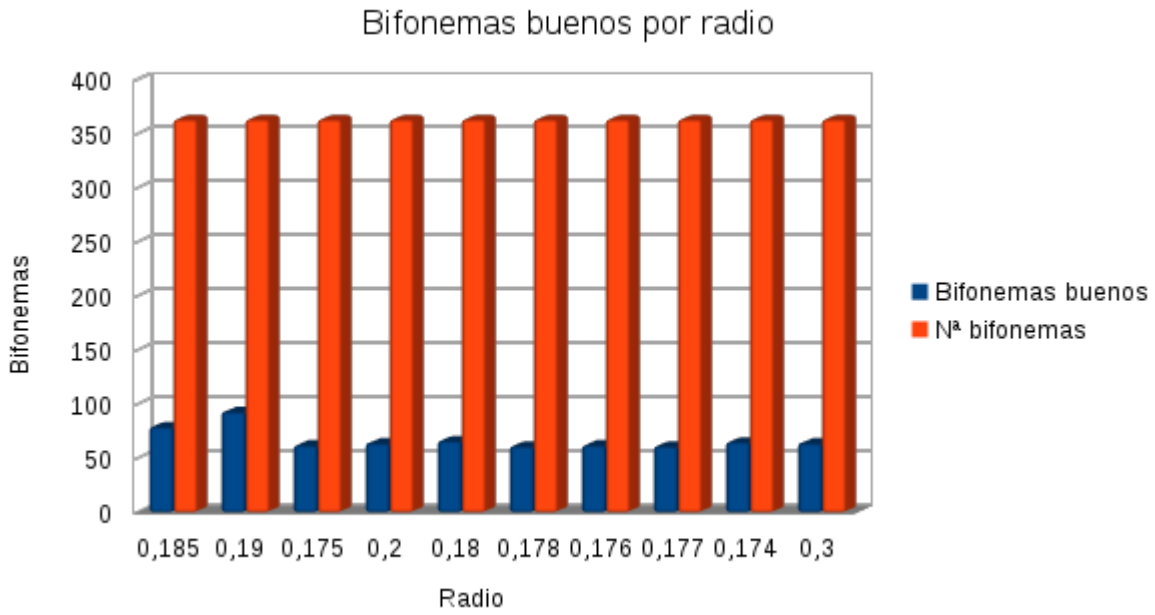

Ilustración 21: Bifonemas buenos por radio

Por otra parte, la Ilustración 21 representa el número de bifonemas con menos de 4 burbujas asociadas, con respecto al total. Resaltan 2 radios por encima de los demás: 0.185 y 0.19, que se comparan en la Ilustración 22 con el peor de los casos.

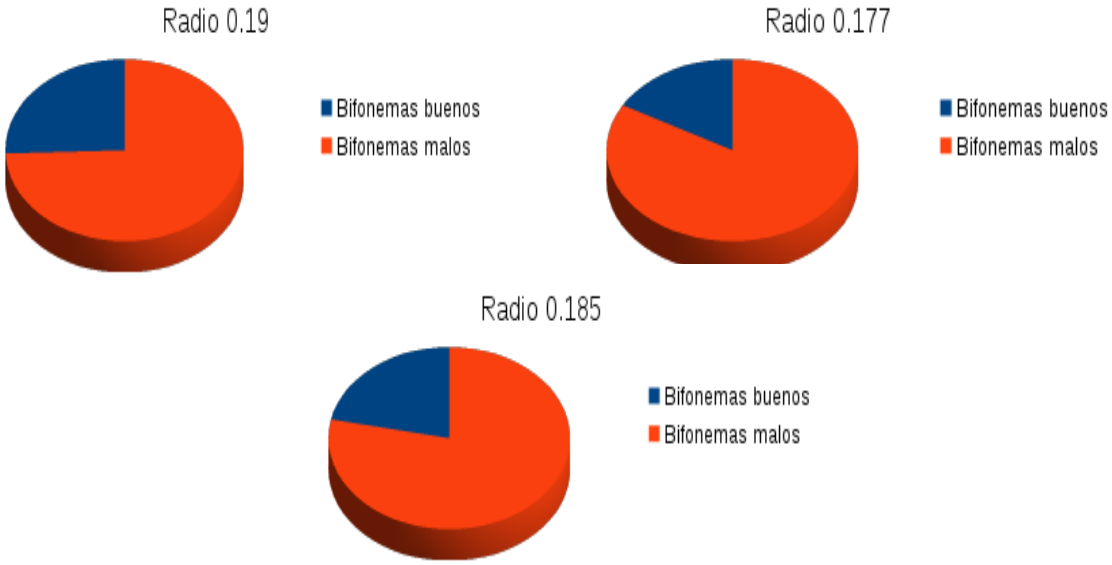

Ilustración 22: Bifonemas buenos, casos destacados

Ambos radios tienen alrededor del 25% de bifonemas buenos, en contraste con algo más del 17% del radio con valor 0.177.

Los datos relativos a los bifonemas buenos son una prueba de que la investigación puede continuar con éxito. Como muestra de las posibilidades que esta vía de análisis tiene, la Ilustración 23 recoge el número de experimentos que extraen cuatro burbujas o menos del mismo bifonema a partir de los datos de la Tabla 7.

Ilustración 23: Bifonemas buenos en común

En la Ilustración 23 se puede observar cómo hay un considerable número de bifonemas que son considerados como “buenos” en 5 o más experimentos.

5.4 Medidas de tiempo

En cuanto al rendimiento del algoritmo, la mayor parte del tiempo de cálculo se emplea en el cálculo de los potenciales. En una máquina de 3’02 GHz de frecuencia de reloj en la CPU con 8 núcleos dedicados, 8 GB de memoria RAM, y con el radio que más grande hacía este cálculo, alcanza las 4 horas. Las siguientes fases del algoritmo son más rápidas, devolviendo resultados después de un intervalo de cinco o diez minutos dependiendo del número de clusters detectados.

6 Conclusiones y trabajos futuros

6.1 Conclusiones técnicas

El objetivo del proyecto era responder a la pregunta de si existen regiones de alta densidad en el espacio de coeficientes cepstrales asociados a tramas de audio de locutores humanos.

Como se ha indicado en la sección 5, las pruebas indican que los vectores de dicho espacio no se distribuyen uniformemente sobre él, sino que existen agrupaciones entorno a puntos de alta densidad que sugieren la posibilidad de que pueda establecerse una correspondencia unívoca entre las unidades acústicas de los audios (para este caso en concreto, bifonemas) y las burbujas resultantes del proceso de clusterización.

Estos resultados confirman que el aprendizaje automático puede desempeñar un rol importante en el descubrimiento y/o mayor comprensión de los procesos propios de la naturaleza humana tales como el habla, el canto, etc.

Asimismo, el hecho de que estos resultados hayan sido obtenidos por un algoritmo de clasificación no supervisado y determinista significa que es posible que la naturaleza de fenómenos aparentemente complejos como el del habla sea mucho más sencilla, ya que los fundamentos de algoritmos como el empleado aquí (Subtractive Clustering) no escapan al alcance del entendimiento de una persona común.

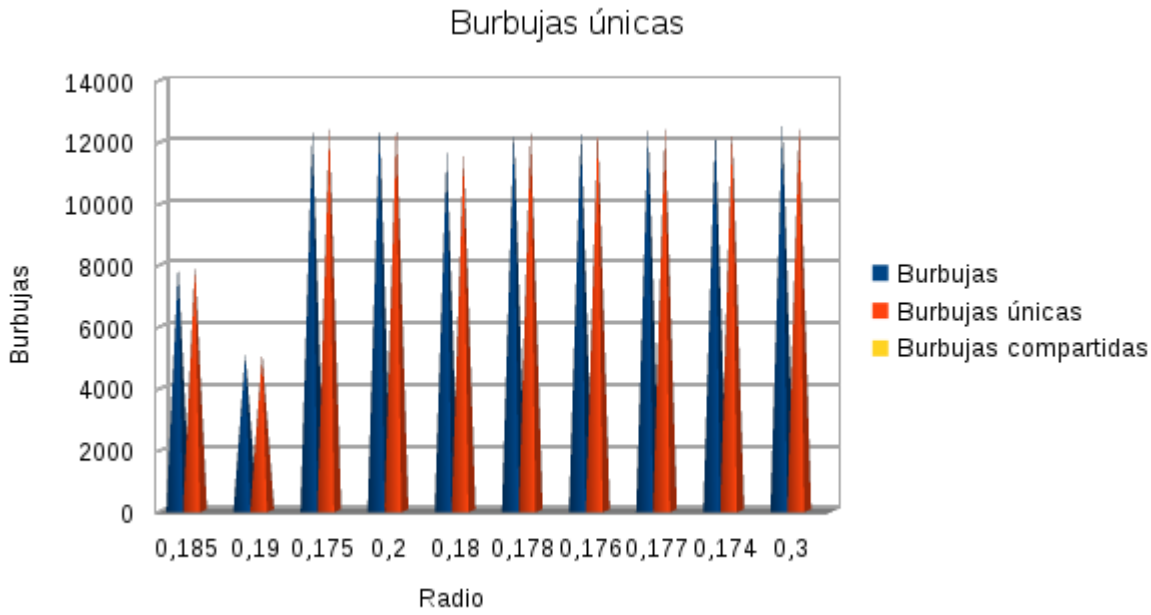

Ilustración 24: Burbujas únicas

Por otra parte, los datos que aparecen en la Tabla 3, especialmente los relativos a la Ilustración 24 y a la Ilustración 22 indican que la investigación de que este proyecto forma parte puede tener resultados positivos. El gran número de bifonemas caracterizados por 4 o menos burbujas, a pesar de que el número total de éstas sugiera que es necesario realizar experimentos con agrupaciones de burbujas (Ilustración 20), es una muestra de que puede existir una caracterización determinista de muchos bifonemas (ver Ilustración 23).

El hecho de que exista una relación uno-a-varios entre los clusters y los bifonemas (Ilustración 24) es sorprendente, ya que algunas de las coarticulaciones se deberían de parecer aunque pertenecieran a bifonemas diferentes. Los datos presentados en estas pruebas no refutan tal afirmación, sino que apoyan la conclusión de que es necesario agrupar las burbujas para disminuir su cantidad total y a su vez provocar que aquellos bifonemas más parecidos entre ellos compartan al menos un porcentaje determinado.

Es por ello que los resultados aquí presentados apuntan a que es necesario continuar con

esta línea de investigación, y que es posible que su hipótesis sea válida y por tanto se puedan caracterizar bifonemas unívocamente a través de sus características MFCC.

6.2 Trabajos futuros

Podrían plantearse al menos 3 proyectos diferentes que se basan en los resultados obtenidos en este.

Por un lado, para resolver el problema del alto número de burbujas, sería necesario plantear una estrategia de unificación que podría consistir en la aplicación del mismo algoritmo de clustering a este nuevo conjunto de datos que constituirían las burbujas actuales.

Por otro lado, para comprobar la viabilidad de estos resultados a un nivel práctico, sería necesario utilizar varios conjuntos de test que comprobaran que las funciones de distribución que constituyen las burbujas de los bifonemas buenos (aquellos caracterizables por un número pequeño de clusters) representan un número significativo de sucesos definidos en los bifonemas correspondientes de los conjuntos de test.

Finalmente, y si los dos proyectos anteriores tuvieran éxito, se podría plantear un proyecto de desarrollo que implementara modelos HMM con burbujas, en lugar de con vectores características directamente.

6.3 Conclusiones personales

La realización de este proyecto ha supuesto un reto para el proyectando dada la naturaleza investigadora del mismo y el comienzo de la vida laboral durante su realización.

Para casi cualquier estudiante de ingeniería, todo lo relativo a Investigación le llama la atención y su imaginación se dispara al considerar el impacto que sus conocimientos y su dedicación pueden tener en el mundo real. Sin embargo, no siempre se opta por explorar este camino al disponer de posibilidades más provechosas a corto plazo (como acabar la carrera antes, o aprovechar las proyectos laborales para reducir la carga de trabajo); en mi

caso, decidí acabar este proyecto porque quería saber qué significa investigar de verdad, qué supone tener trabajo y a la vez necesitar saciar otros intereses no relacionados con él, qué puedo hacer yo ante los retos del estado del arte de un área de la que aún queda mucho por explorar.

Ha supuesto un gran esfuerzo, mucha dedicación y sobre todo mucha fuerza de voluntad. Pero ahora sé que investigar no es tener la idea revolucionaria, sino probar que realmente lo es; que no es esperar a que otros realicen las tareas más aburridas que una tesis implica, sino dedicar el 80% del tiempo a ellas; que no consiste en entender los problemas a la primera, sino que implica rumiarlos y probarlos hasta conseguir atisbar un poco más de su naturaleza.

Gracias a esto, puedo concluir que la experiencia ha sido dura pero valiosa. El tiempo dedicado no ha sido en vano, y aunque los frutos técnicos no han alcanzado (por poco) el grado de madurez que inicialmente esperaba, sí puedo afirmar que serán útiles para trabajos futuros relacionados con el problema de este proyecto.

7 Presupuesto

En esta sección se expondrán los costes de este proceso de experimentación y pruebas.

La Tabla 4 muestra los gastos del proyecto relativos a recursos humanos, en euros.

Tabla 4: Gastos en recursos humanos

Nombre y Apellidos	Precio en €/hora	Dedicación horas/mes	Meses de duración	Importe
Rafael Gálvez	20	60	24	28800
Luis Puente	50	20	24	24000
Total				**52800€**

Los gastos en euros dedicados a la compra de material se desglosan en la Tabla 5.

Tabla 5: Gastos en material

Concepto	Coste	% Uso	Dedicación en meses	Periodo de depreciación	Coste imputable
Licencia Matlab r2012b	90	100	20	60	30
Fungibles	20	100	20	60	6.66
Computadora portátil	1000	50	20	60	166.66
Computadora sobremesa	1200	50	20	60	200
Total					**403.32€**

Para calcular la amortización, se ha utilizado la siguiente fórmula:

$$\frac{Dedicación}{Depreciación} \times Coste \times Uso$$

Los gastos totales se muestran en la Tabla 6.

Tabla 6: Gastos totales

Concepto	Coste
Material	403.32
Recursos humanos	52800
Total sin IVA	53203.32€
Total con IVA	**74107.92€**

8 Trabajos citados

Ambikairajah, E., Kua, J., Sethu, V., & Li, H. (Dec de 2012). PNCC-ivector-SRC based speaker verification. *Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, (págs. 1-7).

Aparicio, R. K., & Acuña, E. (2009). Clasificación Semi-Supervisada de Documentos.

Ayyasamy, R. K. (2013). Organizing Information in the Blogosphere: The Use of Unsupervised Approach.

Ball, G. H., & Hall, D. J. (1965). *ISODATA, a novel method of data analysis and pattern classification.* Tech. rep., DTIC Document.

Bansal, P., Kant, A., Kumar, S., Sharda, A., & Gupta, S. (2008). Improved hybrid model of HMM/GMM for speech recognition.

Benzeghiba, M., De Mori, R., Deroo, O., Dupont, S., Erbes, T., Jouvet, D. et al.. (2007). Automatic speech recognition and speech variability: A review. *Speech Communication, 49*(10), 763-786.

Chauhan, T., Soni, H., & Zafar, S. (2013). A review of automatic. *International Journal of Soft Computing and*.

Chiu, S. L. (1994). Fuzzy model identification based on cluster estimation. *Journal of intelligent and Fuzzy systems, 2*(3), 267-278.

de Kok, D., & Brouwer, H. (2011). Natural language processing for the working programmer. *Natural language processing for the working programmer*. Del.

De Lacy, P. V. (2007). *The Cambridge handbook of phonology.* Cambridge University Press.

Drineas, P., Frieze, A., Kannan, R., Vempala, S., & Vinay, V. (2004). Clustering large graphs via the singular value decomposition. *Machine learning, 56*(1-3), 9-33.

Espinoza, F. M., et al.. (2012). Identificación de hablantes a partir de trayectorias

temporales en unidades lingúísticas sobre grandes bases de datos.

Etxebarria, M. (2013). Iniciación a la fonética acústica. *Anuario del Seminario de Filología Vasca" Julio de Urquijo", 21*(2), 475-514.

Fernández, A. M. (2012). Aspectos fonéticos del proceso de velarización en las nasales del español y del catalán.

Ferrán, J. M. (2004). Selección diferenciada del conjunto de entrenamiento en redes de neuronas mediante aprendizaje retardado.

Forcada, V. R. (2003). *Clasificación supervisada basada en redes bayesianas, aplicación en biología computacional.* Ph.D. dissertation, Universidad Politécnica de Madrid.

Fustes, D. (2014). Extracción de conocimiento en bases de datos astronómicas mediante redes de neuronas artificiales: aplicaciones en la misión Gaia.

García, J. G. (2014). *Sistema de reconocimiento de voz usando perceptrón multicapa y Coeficientes Cepstrales de Mel.* Ph.D. dissertation.

Gordon, A. D. (1999). Classification, (Chapman \& Hall/CRC Monographs on Statistics \& Applied Probability).

Gu, G., Perdisci, R., Zhang, J., Lee, W., et al.. (2008). BotMiner: Clustering Analysis of Network Traffic for Protocol-and Structure-Independent Botnet Detection. *USENIX Security Symposium*, (págs. 139-154).

Han, J., & Kamber, M. (2011). Data mining concepts and techniques 3rd.

Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer, 27*(2), 83-85.

Haykin, S. (1999). Neural networks. *A Comprehensive Foundation*.

Hebb, D. (1968). 0.(1949) The organization of behavior. *0.(1949) The organization of behavior*. Wiley, New York.

Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *the Journal of*

the Acoustical Society of America, 87(4), 1738-1752.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N. et al.. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE, 29*(6), 82-97.

Hualde, J. I. (2005). *The sounds of Spanish.* Cambridge University Press New York.

Inal, M. (2007). Feature Extraction of Speech Signal by Genetic Algorithms-Simulated Annealing and Comparison with Linear Predictive Coding Based Methods. En *Adaptive and Natural Computing Algorithms* (págs. 266-275). Springer.

Iqbal, S., Mahboob, T., & Khiyal, M. S. (2011). Voice Recognition using HMM with MFCC for Secure ATM. *International Journal of Computer Science Issues (IJCSI), 8*(6).

Isasi, P. (1996). Modelos neuronales competitivos: Kohonen \& ART.

Isasi, P., & Galván, I. M. (2004). *Redes de neuronas artificiales: un enfoque práctico.* Prentice Hall.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters, 31*(8), 651-666.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data.* Prentice-Hall, Inc.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR), 31*(3), 264-323.

Jain, R. (2012). A hybrid clustering algorithm for data mining. *arXiv preprint arXiv:1205.5353*.

Kamaruddin, N., & Wahab, A. (2008). Speech Emotion Verification System (SEVS) based on MFCC for real time applications.

Kass, R. E., & Wasserman, L. (1995). A reference Bayesian test for nested hypotheses and its relationship to the Schwarz criterion. *Journal of the american statistical association, 90*(431), 928-934.

Kim, C., & Stern, R. M. (2009). Feature extraction for robust speech recognition using a power-law nonlinearity and power-bias subtraction. *INTERSPEECH*, (págs. 28-31).

Kim, C., & Stern, R. M. (2012). Power-normalized cepstral coefficients (PNCC) for robust speech recognition. *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, (págs. 4101-4104).

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE, 78*(9), 1464-1480.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Supervised machine learning: A review of classification techniques*.

Kumar, P., Jakhanwal, N., Bhowmick, A., & Chandra, M. (2011). Gender classification using pitch and formants. *Proceedings of the 2011 International Conference on Communication, Computing \& Security*, (págs. 319-324).

Ladefoged, P., & Johnson, K. (2014). *A course in phonetics.* Cengage learning.

Larrañaga, P., Inza, I., & Abdelmalik, M. (2008). Tema 14. Clustering. *Tema 14. Clustering*.

Lee, K.-F., Hayamizu, S., Hon, H.-W., Huang, C., Swartz, J., & Weide, R. (1990). Allophone clustering for continuous speech recognition. *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, (págs. 749-752).

Li, B., & Sim, K. C. (2014). A Spectral Masking Approach to Noise-Robust Speech Recognition Using Deep Neural Networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22*(8), 1296-1305.

Li, Q., He, Y., & Jiang, J.-p. (2008). A New Clustering Algorithm Based Upon Flocking On Complex Network. *arXiv preprint arXiv:0812.5032*.

Llisterri, J., Aguilar, L., Garrido, J. M., Machuca, M., Marín, R., de la Mota, C. et al.. (1999). Fonética y tecnologías del habla. *Filología e Informática. Nuevas tecnologías en los estudios filológicos*, 449-479.

Lloyd, S. (1982). Least squares quantization in PCM. *Information Theory, IEEE*

Transactions on, 28(2), 129-137.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, *1*, págs. 281-297.

Meilua, M. (2006). The uniqueness of a good optimum for k-means. *Proceedings of the 23rd international conference on Machine learning*, (págs. 625-632).

Mirkin, B. (2012). *Clustering: a data recovery approach.* CRC Press.

Mitterer, H., Scharenborg, O., & McQueen, J. M. (2013). Phonological abstraction without phonemes in speech perception. *Cognition, 129*(2), 356-361.

Moreno, C. P. (2002). *Reconocimiento de habla mediante transparametrización: una alternativa robusta para entornos móviles e IP.* Ph.D. dissertation, Universidad Carlos III de Madrid.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (1998). Learning to classify text from labeled and unlabeled documents. *AAAI/IAAI, 792*.

Obediente, E. (1998). *Fonética y fonología.* Universidad Los Andes.

Pelleg, D., Moore, A. W., et al.. (2000). X-means: Extending K-means with Efficient Estimation of the Number of Clusters. *ICML*, (págs. 727-734).

Planas, A. M. (2012). Aspectos fonéticos del proceso de velarización en las nasales del español y del catalán. *Onomázein, 2*(26), 95-129.

Puente, L. (2014). Influencia de los segmentos del discurso en la discriminación del locutor.

Puente, L., Poza, M. J., Ruíz, B., & Carrero, D. (2011). Biometrical Fusion--Input Statistical Distribution.

Rabiner, L., Juang, B.-H., & Lee, C.-H. (1996). An overview of automatic speech recognition. En *Automatic Speech and Speaker Recognition* (págs. 1-30). Springer.

Rumsey, F., & McCormick, T. (2014). *Sound and Recording: Applications and Theory.* CRC Press.

Sárosi, G., Mozsáry, M., Mihajlik, P., & Fegyó, T. (2011). Comparison of feature extraction methods for speech recognition in noise-free and in traffic noise environment. *Speech Technology and Human-Computer Dialogue (SpeD), 2011 6th Conference on*, (págs. 1-8).

Solera, R. (2011). *Máquinas de vectores soporte para reconocimiento robusto de habla.* Ph.D. dissertation, Universidad Carlos III de Madrid.

Steinbach, M., Karypis, G., Kumar, V., et al.. (2000). A comparison of document clustering techniques. *KDD workshop on text mining*, *400*, págs. 525-526.

Steinhaus, H. (1956). Sur la division des corp materiels en parties. *Bull. Acad. Polon. Sci, 1*, 801-804.

Stevens, K. N. (2007). Models of speech production. *Encyclopedia of Acoustics, Volume Four*, 1565-1578.

Torres, B., Antonio, J., & Casado, J. (2007). Anatomía funcional de la voz. *Medicina del canto. Capítulo, 1*.

Valero, X., & Alías, F. (2012). Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification. *Multimedia, IEEE Transactions on, 14*(6), 1684-1689.

Vidal, E. (2014). Algoritmo divisivo de clustering con determinación automática de componentes.

Viele, K. (2008). Multivariate Normal Distribution. *Multivariate Normal Distribution*.

Walker, W., Lamere, P., Kwok, P., Raj, B., Singh, R., Gouvea, E. et al.. (2004). Sphinx-4: A flexible open source framework for speech recognition.

Wu, J.-D., & Lin, B.-F. (2009). Speaker identification based on the frame linear predictive coding spectrum technique. *Expert Systems with Applications, 36*(4), 8056-8063.

Xiang, S., Nie, F., & Zhang, C. (2008). Learning a Mahalanobis distance metric for data clustering and classification. *Pattern Recognition, 41*(12), 3600-3612.

Xie, F., Fan, S., Wang, J., Lu, H., & Li, C. (2014). Artificial Intelligence and Data Mining 2014. *Abstract and Applied Analysis*, *2014*.

Yager, R. R., & Filev, D. P. (1994). Approximate clustering via the mountain method. *Systems, Man and Cybernetics, IEEE Transactions on, 24*(8), 1279-1284.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X. et al.. (1997). *The HTK book* (Vol. 2). Entropic Cambridge Research Laboratory Cambridge.

Zatorre, R. (2005). Music, the food of neuroscience? *Nature, 434*(7031), 312-315.

Zubiaga, A., Fresno, V., & Martínez, R. (2009). Comparativa de aproximaciones a SVM semisupervisado multiclase para clasificación de páginas Web. *María Teresa Vicente-Díez, Paloma Martínez, Ángel Martínez-González, 42*, 63-70.

9 Apéndice A: Tabla de bifonemas buenos

Como se definió en la sección de resultados, un bifonema bueno es aquel del cual se extraen cuatro o menos burbujas diferentes.

La Tabla 7 que se muestra en este apéndice indica cuántos experimentos han clasificados los mismos bifonemas como buenos, y a partir de ella se compone la Ilustración 22.

Tabla 7: Bifonemas buenos repetidos en experimentos

Bifonema bueno	Nº Experimentos
>>	10
aa	5
ag	1
aN	1
ao	2
av	10
>b	9
bo	1
br	1
bs	8
bt	10
bu	8

ci	6
dd	6
dk	2
dp	9
dz	10
e~	1
ec	10
ee	9
eh	3
eL	10
eY	9
>f	9
>g	9
ge	10
gi	2
gn	2
hu	10
<i	9
i<	9

i<	9
ii	9
ip	8
iy	2
ja	1
ju	10
<k	8
kn	10
kr	2
ks	1
ky	10
La	3
lb	1
lg	10
lm	1
ln	4
lR	9
ls	4
lt	1

mb	1
md	10
nb	2
nn	8
nX	1
o>	4
og	1
oj	1
oo	10
ou	2
ov	8
oX	3
oy	8
pt	1
r<	10
>r	9
Ra	10
rb	10
Rb	2

Rc	10
Rf	4
Rg	9
Rk	1
Rl	1
rl	10
rn	10
Rp	1
rp	9
rs	8
rt	2
Ru	10
ru	7
RX	10
Rz	1
rz	3
s>	2
sf	1
sg	9

sj	10
sR	10
sz	1
>t	1
tn	10
tp	10
u~	5
ub	9
ud	1
ui	1
uj	2
uk	10
up	8
uR	2
uX	1
ux	4
uy	2
uz	1
va	9

ve	2
Xa	1
xa	10
Xe	2
Xi	10
>y	7
ye	1
yi	9
yu	1
>z	1
z<	3
zm	10
zo	1
zp	10
zs	1
zu	1
zz	6