Universidad Nacional Autónoma de México

Programa de Maestría y Doctorado en Ingeniería

RECONOCIMIENTO DE PALABRAS CLAVE EN TEXTO HABLADO

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

MAESTRO EN INGENIERÍA

INGENIERÍA ELÉCTRICA
PROCESAMIENTO DIGITAL DE SEÑALES

P R E S E N T A:

ROY BALDERAS JIMÉNEZ

T U T O R:

DR. JOSÉ ABEL HERRERA CAMACHO

2012

Jurado asignado:

Presidente:	Dr. Orduña Bustamante Felipe
Vocal:	Dr. Escalante Ramírez Boris
Secretario:	Dr. Herrera Camacho José Abel
Primer suplente:	Dr. Sierra Martinez Gerardo Eugenio
Segundo suplente:	Dr. Medina Urrea Alfonso

Facultad de Ingeniería, UNAM. México D.F.

Tutor de tesis:

Dr. José Abel Herrera Camacho

Firma

A la amada memoria de mi Padre,

Ramón Balderas Sánchez.

Agradecimientos

Agradezco a la Universidad Nacional Autónoma de México por nuevamente concederme el alto honor de formar parte de su valiosa comunidad estudiantil, lo cual me llena de orgullo.

Quiero expresar mi más sincero agradecimiento al Doctor José Abel Herrera Camacho por brindarme su invaluable guía, apoyo, tiempo y paciencia a lo largo de la elaboración del presente trabajo de investigación.

Agradezco a los miembros del jurado, los Doctores Felipe Orduña Bustamante, Boris Escalante Ramírez, Gerardo Eugenio Sierra Martinez, Alfonso Medina Urrea y José Abel Herrera Camacho por su valioso apoyo, sugerencias y comentarios, que no sólo enriquecieron el presente trabajo de investigación sino que también me permitieron reevaluarlo con nuevas y originales perspectivas.

Agradezco enormemente al Consejo Nacional de Ciencia y Tecnología (CONACYT) el apoyo otorgado a través de una beca CONACYT-Nacional para estudios de posgrado de la cual fui beneficiario, pero sobre todo por haber sido desde su creación un factor determinante para el desarrollo de la ciencia en México.

Al Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT) ya que parte de este trabajo fue financiado por él, a través uno de sus proyectos. A la entidad académica responsable del proyecto, el Instituto de Ingeniería y muy especialmente a los Doctores Alfonso Medina Urrea y José Abel Herrera Camacho responsables del proyecto y quienes me dieron la oportunidad de participar en él.

Agradezco al Posgrado de Ingeniería y muy particularmente a los profesores del departamento de procesamiento digital de señales quienes con su dedicación y esfuerzo me ofrecieron la mejor formación académica posible en el área de especialidad. A todos ellos mi más cumplido reconocimiento.

Quisiera en esta oportunidad recordar y agradecer a todos los compañeros y colegas, amigos todos, por los importantes momentos que compartimos durante los estudios de maestría y la elaboración del presente trabajo de investigación. Particularmente a Francisco Javier Ayala, Benjamín Gutierrez, Anibal Orozco, Jaime Alfonso Reyes, Marlene Rodríguez, Abundio Rodríguez y Andrés Villarreal.

Quiero expresar mi más profundo agradecimiento a mi familia y muy especialmente a mi madre por insistir siempre en la importancia de una buena educación, por su fuerza, confianza y apoyo, que de forma permanente han sido parte fundamental para lograr metas cada vez más elevadas.

A Mariza mi más grande agradecimiento, por su amor, ternura y paciencia y mi enorme admiración por ser fuente de inspiración y ejemplo permanente de superación académica.

Índice

Índice de figuras

Índice de Tablas

Capítulo 1
Introducción

En la era de la información, las aplicaciones informáticas se han convertido en parte de la vida moderna y esto a su vez ha alentado las expectativas de una interacción amigable y natural con ellas. El habla, como medio de comunicación, ha observado el desarrollo exitoso de un gran número de aplicaciones de reconocimiento automático del habla (RAH), incluyendo el reconocimiento de comandos y control, los sistemas de diálogo para personas con capacidades diferentes, la traducción, el dictado, etc.

Pero el problema real va más allá del uso del habla en aplicaciones de control o de acceso a la información. El objetivo es utilizar el habla como fuente de información, compitiendo por ejemplo, con el documento de texto en línea. Dado que la tecnología que da soporte a dichas aplicaciones informáticas depende en gran medida del desempeño de los sistemas de RAH, la investigación en esta área sigue siendo un tema con gran actividad, como lo demuestra la amplia gama de líneas de investigación sugeridas en Baker et al. [2009].

El reconocimiento automático del habla entendido como el reconocimiento de la información inherente en una señal de voz y su transcripción en términos del conjunto de caracteres correspondiente [Junqua y Haton, 1995] ha sido objeto de intensa investigación durante más de cinco décadas, logrando resultados notables. Es de esperar que los avances en el reconocimiento automático del lenguaje hablado permitan que sea tan conveniente y accesible como el texto en línea, cuando los reconocedores alcancen tasas de error cercanas a cero. Pero en tanto que el reconocimiento de dígitos ya ha alcan-

zado una tasa del 99,6 % [Lee et al., 2007], no se puede afirmar lo mismo del reconocimiento de fonemas, para el cual la tasas más altas siguen siendo inferiores al 80 % [Mohamed et al., 2011; Siniscalchi et al., 2007].

El reconocimiento del habla basado en fonemas es muy atractivo ya que está inherentemente libre de las limitaciones que impone el vocabulario. El desempeño de los sistemas de RAH con vocabularios grandes (comúnmente conocidos como sistemas LVASR[1]) dependen de la calidad del reconocedor de fonemas. Es por eso que un número importante de grupos de investigación continúan desarrollando este tipo de reconocedores, con el objetivo de mejorar su desempeño tanto como sea posible. Este tipo de reconocimiento es de hecho un problema recurrente para la comunidad dedicada al reconocimiento automático del habla.

Existe una amplia gama de aplicaciones para el reconocimiento de fonemas, además de los típicos sistemas LVASR [Scanlon et al., 2007; Morris y Fosler-Lussier, 2008] también se encuentra en aplicaciones de detección o reconocimiento de palabras clave, que son materia del presente trabajo y de las cuales hablaremos ampliamente, en el reconocimiento del idioma [Matějka, 2008], en la identificación y verificación del hablante [Furui, 2005] y en aplicaciones para identificación de música [Fujihara y Goto, 2008] por citar sólo algunas.

El desafío de construir modelos acústicos robustos implica la aplicación de buenos algoritmos de aprendizaje al conjunto de datos adecuado. La base de datos con grabaciones de audio de expresiones habladas y sus transcripciones, o bien *corpus del habla* define las unidades que pueden ser aprendidas y el éxito de los algoritmos de aprendizaje depende en gran medida de la calidad y el detalle de las anotaciones que se hagan de esas unidades en dicho corpus. Muchas bases de datos no cuentan con suficientes anotaciones y sólo unas cuantas incluyen etiquetas a nivel fonético. Así que la razón principal por la cual la base de datos TIMIT [Garofolo et al., 1993] se ha convertido en el corpus del habla más utilizado por la comunidad de investigadores del reconocimiento de fonemas se debe principalmente a que está etiquetada manualmente en su totalidad a nivel fonético. El reconocimiento de fonemas usando el corpus TIMIT tiene más de dos décadas de intensa investigación y naturalmente su desempeño ha mejorado con el tiempo.

[1]Por sus siglas en inglés *Large Vocabulary Automatic Speech Recognition*.

Existe una amplia variedad de sistemas de reconocimiento de diferentes tipos, pero en lo referente a la evaluación de su desempeño se centra principalmente en tres tareas: la segmentación, la clasificación y el reconocimiento de fonemas. Mientras que la primera alcanza tasas del 93 % [Hosom, 2009], la segunda llega al 83 % [Karsmakers et al., 2007]. Y la tercera se mantiene en aproximadamente 79 % [Mohamed et al., 2011; Siniscalchi et al., 2007].

La segmentación de la señal de voz a nivel fonético es el proceso de búsqueda de las fronteras o límites de una determinada secuencia de fonemas conocidos en un lenguaje hablado. Esta delimitación de las fronteras a nivel fonético representa un problema complejo debido a los efectos de la coarticulación, en la cual los fonos adyacentes se influyen mutuamente. La clasificación fonética es un problema artificial, pero aleccionador en el área del RAH [Sha y Saul, 2006]. En este problema tomamos la señal correctamente segmentada, pero cuya etiqueta correspondiente para cada segmento es desconocida. El problema consiste en identificar correctamente los fonemas en esos segmentos. Los modelos de fonos compiten entre si unos contra otros en un intento por elegir la etiqueta adecuada en el segmento respectivo. La etiqueta del modelo ganador se compara con la etiqueta del corpus TIMIT correspondiente y se establece una ocurrencia de acierto o de error. La clasificación de fonemas, permite evaluar la calidad del modelado acústico, ya que calcula el desempeño del reconocedor sin el uso de algún tipo de gramática [Reynolds y Antoniou, 2003].

El reconocimiento de fonemas obedece a criterios más severos y complejos. La expresión hablada dada al reconocedor corresponde a una expresión completa. Comúnmente los modelos de fonos junto con una decodificación de Viterbi bastan para encontrar la mejor secuencia de etiquetas para la expresión hablada de entrada y en este caso puede utilizarse una gramática. La mejor secuencia de fonemas, encontrada por el algoritmo de Viterbi se compara con la referencia (las etiquetas establecidas manualmente en el corpus TIMIT para el mismo enunciado), utilizando un algoritmo de programación dinámica, usualmente la distancia Levenshtein (o distancia de edición), la cual toma en cuenta los aciertos en el etiquetado de fonemas, sustituciones, eliminaciones e inserciones.

El uso de modelos ocultos de Markov (MOM) para aplicaciones de reconocimiento del habla es muy amplio, al menos para el modelado de eventos

en el tiempo. Después de décadas de intensa investigación todo indica que el desempeño de los sistemas RAH basados en MOM ha alcanzado gran estabilidad. A finales de la década de los 80´s las redes neuronales artificiales (RNA) reaparecieron como una alternativa a los MOM. Aparecieron métodos híbridos MOM/RNA que han logrado resultados comparables y a veces superiores a aquellos basados en MOM. En la última década sin embargo, han aparecido dos nuevas técnicas en el campo del aprendizaje automático, con resultados sorprendentes para las tareas de clasificación: la primera de ellas, las máquinas de vectores de soporte (MVS), cuya posible utilidad en el reconocimiento automático del habla ha inspirado el desarrollo del presente trabajo. La segunda y más reciente son los campos aleatorios condicionales (CAC). Partimos del hecho de que hasta ahora los mejores resultados usando el corpus TIMIT se han alcanzado con el uso de modelos híbridos MOM/RNA [Rose y Momayyez, 2007; Scanlon et al., 2007; Siniscalchi et al., 2007], y de modelos MOM/CAC, también híbridos [Morris y Fosler-Lussier, 2008].

1.1. Definición del problema y objetivos

El propósito de este trabajo es presentar una aproximación diferente al reconocimiento automático del habla que no está basada en *modelos generativos* tales como los modelos ocultos de Markov sino en los últimos avances en métodos núcleo y de margen amplio que son considerados *enfoques discriminatorios*. Dividimos la enorme tarea del reconocimiento del habla en pequeñas tareas manejables y descritas anteriormente: la segmentación, la clasificación y el reconocimiento de fonemas; en donde la solución implementada para cada una de ellas demuestra la utilidad del método propuesto.

Al final del trabajo de investigación aplicamos las tareas descritas mediante la implementación de un detector de palabras clave contenidas dentro de expresiones del habla continua y examinamos la pertinencia del uso de este método discriminatorio para tal detección.

Dada una palabra representada con una secuencia de fonemas, y una expresión hablada, el reconocedor de palabras clave debe ser capaz de predecir el intervalo de tiempo en que ocurre la secuencia de fonemas dentro de la expresión, junto con un nivel de confianza respecto de dicha predicción. Si la

predicción del valor de confianza está por encima de cierto nivel se declara que la palabra clave se pronunció en la expresión en el intervalo de tiempo predicho.

El problema del entrenamiento para el reconocimiento de palabras clave se formula como una tarea discriminatoria, donde los parámetros del modelo se eligen de manera que la expresión en la que se realiza la palabra clave tendría un mayor nivel de confianza que en cualquier otra expresión hablada en donde no existe una realización de dicha palabra.

Puesto que la medida más común para evaluar reconocedores de palabras clave es el área bajo la *curva de características operativas del receptor* (mejor conocida como curva ROC[2]). Se propone mostrar teórica y empíricamente que de la aplicación del método de entrenamiento discriminatorio propuesto resulta una gran área bajo la curva ROC.

Para ello es necesaria la creación y presentación de un algoritmo iterativo para entrenar eficientemente al detector de palabras clave. La aproximación sugerida contrasta con las estrategias estándar de reconocimiento basadas en modelos ocultos de Markov (MOM), para las cuales el procedimiento de entrenamiento no maximiza la pérdida directamente relacionada con el desempeño del detector de palabras. Resulta pues necesaria la realización de experimentos utilizando un corpus estándar del habla que permita observar las ventajas de la aproximación sugerida, sobre las alternativas basadas en MOM, y para ello utilizamos el corpus fonético-acústico del habla continua TIMIT. Las razones por las cuales se realizó dicha elección se expondrán con mayor detalle a lo largo del trabajo[3].

1.2. Metodología

Abordamos el estudio y utilización de cada una de las tareas enunciadas de la siguiente manera, primero definimos la noción de error para la tarea, para luego diseñar un algoritmo discriminatorio y ofrecer un análisis que

[2]Por sus siglas en inglés Receiver Operating Characteristic.

[3]Algunas de ellas ya se han presentado más arriba, pero una exposición detallada sobre la estructura y características del corpus TIMIT (elegido para elaborar los experimentos que demuestran la utilidad del método propuesto) se presenta en el *Apéndice A*.

demuestra que el algoritmo de hecho reduce al mínimo este error tanto en el conjunto de datos de entrenamiento como en el de pruebas.

Comenzamos describiendo un sistema típico de reconocimiento del habla y formulamos sus limitaciones e inconvenientes. Como resultado se deduce que el principal problema en los sistemas de reconocimiento del habla es en efecto el modelado acústico. A continuación revisamos los conceptos fundamentales de los métodos núcleo y de margen amplio así como sus ventajas. Posteriormente se describe el marco teórico del modelado acústico, explicando cada una de las tareas necesarias para la construcción de un reconocedor del habla discriminatorio y como se relacionan entre sí.

Finalmente y una vez implementados los algoritmos propuestos para cada una de las tareas enunciadas, se describen los experimentos realizados para comparar el desempeño de los métodos discriminatorios propuestos con los respectivos métodos basados en enfoques generativos MOM y documentados exhaustivamente en la literatura relacionada con el tema.

Capítulo 2
Antecedentes

Resumen:

Una de las herramientas más naturales de comunicación usada por los humanos es su voz. Por lo tanto resulta comprensible que se haya dedicado una gran cantidad de investigación a analizar y entender la pronunciación del habla humana para varias importantes aplicaciones. La más obvia es quizás el *reconocimiento automático del habla*, en donde el objetivo es transcribir una expresión hablada, en su correspondiente secuencia de palabras en forma escrita. Otras aplicaciones incluyen el *reconocimiento del hablante*, en donde el objetivo es determinar quien es la persona que habla (identificación) así como comprobar su identidad declarada (verificación) y la *segmentación del hablante* cuyo propósito es separar una secuencia acústica en relación a sus hablantes, tal como sucede durante un diálogo.

El objetivo de este capítulo es presentar una breve revisión del estado actual del reconocimiento del habla y del enfoque predominante basado en *modelos ocultos de Markov* (MOMs), así como sus limitaciones y problemas conocidos, para después introducir conceptos básicos de los métodos núcleo y de margen amplio, en particular una de sus aproximaciones más y mejor conocidas, las máquinas de vectores de soporte (MVS).

2.1. El enfoque tradicional para el procesamiento del habla

La mayoría de los problemas de procesamiento de voz, incluyendo el Reconocimiento Automático del Habla (RAH), proceden básicamente con el mísmo enfoque, que se describe aquí. La descripción se da en el contexto del RAH, ya que este es el campo que ha atraído la mayor parte de la investigación en los últimos 40 años. Comenzamos con la descripción de un sistema típico de reconocimiento basado en MOM [Young, 1996]. La notación básica se ilustra en la Fig. 2.1.

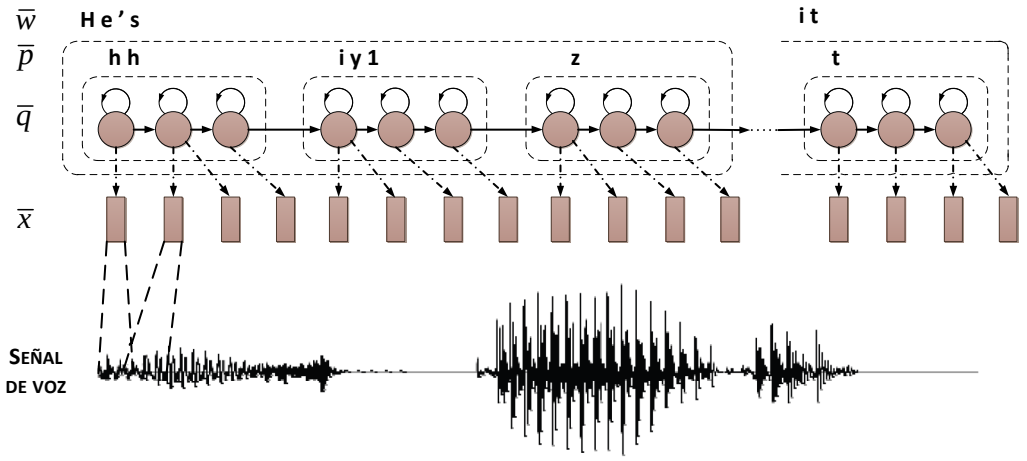

FIGURA 2.1: Notación básica para reconocimiento del habla basado en modelos ocultos de Markov. En donde $\overline{w}$ representa a la *secuencia de palabras*, $\overline{p}$ a la *secuencia de fonemas*, $\overline{q}$ a los *modelos acústicos* y $\overline{x}$ a los *vectores acústicos*.

Dicho enfoque se basa en el siguiente marco estadístico. Una secuencia de vectores de características acústicas se extrae de una expresión hablada por medio de un *front-end* procesador de señales. Denotamos a los vectores de características acústicas como $\overline{\mathbf{x}} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T)$, en donde $\mathbf{x}_t \in \mathcal{X}$ y $\mathcal{X} \subset \mathbb{R}^d$ es el dominio de los vectores acústicos. Cada vector es una representación compacta del espectro de tiempo corto. Típicamente, cada vector cubre un periodo de 10 ms y existen aproximadamente $T = 300$ vectores acústicos en una expresión de 10 palabras.

La expresión hablada consiste en una secuencia de palabras $\overline{v} = (v_1, ..., v_N)$, en donde cada una de ellas pertenece a un vocabulario $\mathcal{V}$ que es fijo y conocido, es decir $v_i \in \mathcal{V}$. La tarea del reconocedor del habla es predecir la secuencia de palabras más probable $\overline{v}'$ dada la señal acústica $\overline{\mathbf{x}}$. El reconocimiento del habla se formula como un problema de decodificación de *Máximo a Posteriori* (MAP) de la siguiente manera

$$\bar{v}' = \underset{\bar{v}}{\operatorname{argmax}} P(\bar{v}|\bar{\mathbf{x}}) = \underset{\bar{v}}{\operatorname{argmax}} \frac{p(\bar{\mathbf{x}}|\bar{v})P(\bar{v})}{p(\bar{\mathbf{x}})}, \tag{2.1}$$

en donde usamos la regla de Bayes para descomponer la probabilidad posterior de la Ec. 2.1. El término $p(\overline{\mathbf{x}}|\overline{v})$ representa la probabilidad de observar la secuencia de vectores acústicos $\overline{\mathbf{x}}$ dada una secuencia de palabras $\overline{v}$ y se le conoce como *el modelo acústico*. El término $P(\overline{v})$ representa la probabilidad de observar una secuencia de palabras $\overline{v}$ y se le conoce como *el modelo de lenguaje*.

La aproximación basada en la *Estimación de la Máxima Verosimilitud* (EMV) que se muestra en la Ec. 2.1, puede también abordarse como un problema clásico de canal ruidoso de comunicación [Rabiner y Juang, 1993]. Esta bien conocida formulación del problema de reconocimiento del habla expresa de forma concisa la relación entre varias fuentes de información usando la regla de Bayes. Dicha formulación es importante por dos razones:

Primero, el término asociado con el modelo acústico $p(\overline{\mathbf{x}}|\overline{v})$ expresa la idea de un proceso mediante el cual los modelos estadísticos pueden ser entrenados. La estimación de la probabilidad de una secuencia de observación $\overline{\mathbf{x}}$ dada una secuencia de palabras $\overline{v}$ puede lograrse mediante la utilización de una base de datos de expresiones habladas (corpus del habla) con la correcta secuencia de palabras, y el uso de un proceso de *Maximización de la Esperanza* (ME) que garantice la convergencia de los parámetros del modelo a una solución del tipo de la Ec. 2.1 para la Estimación de la Máxima Verosimilitud. Segundo, el término $p(\overline{\mathbf{x}})$ en el denominador, que representa la probabilidad de la secuencia de observación, puede ser ignorado dado que es constante durante el proceso de maximización (operación máx). Por lo tanto, el enfoque de EMV para el problema de reconocimiento automático del habla descrito a través de la Ec. 2.1 a menudo se simplifica como

$$\bar{v}' = \underset{\bar{v}}{\operatorname{argmax}}\, p(\bar{\mathbf{x}}|\bar{v})P(\bar{v}), \tag{2.2}$$

Un diagrama de bloques conceptual de esta aproximación se muestra en la Fig. 2.2. Existen cuatro componentes principales del sistema: la *extracción de características*, el *modelado acústico*, el *modelado del lenguaje*, y la *búsqueda*.

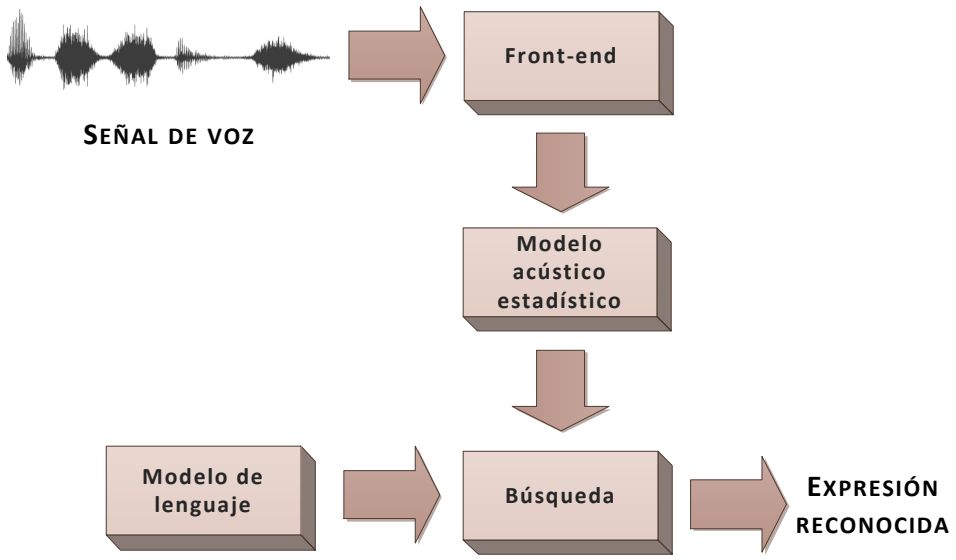

FIGURA 2.2: Los cuatro componentes principales de un sistema de reconocimiento del habla.

El modelo acústico usualmente se estima por medio de un modelo oculto de Markov [Rabiner y Juang, 1993], que es un tipo de modelo que representa la probabilidad conjunta de una variable observable y una oculta [Jelinek, 1998]. La utilidad de las representaciones mediante MOMs reside en su capacidad para modelar la evolución temporal de una señal a través de un proceso oculto de Markov. La capacidad de un MOM para modelar estadísticamente la variabilidad acústica y temporal en el habla ha sido esencial para su éxito.

La distribución de probabilidad asociada a cada estado en un MOM modela la variabilidad que se produce en el habla debido al cambio de hablante o del contexto fonético. Esta distribución es típicamente un *Modelo de Mezclas de Gaussianas* (MMG), ya que este es un modelo paramétrico suficientemente general, así como un marco matemático eficiente y robusto para la estimación y el análisis. El uso generalizado de los MOMs para el modelado del habla se

puede atribuir a la disponibilidad de procedimientos de estimación de parámetros eficientes, tales como la EMV. Una de las razones más convincentes para el éxito de los *estimadores de máxima verosimilitud* y los MOMs ha sido la existencia de métodos iterativos para estimar los parámetros que garantizan la convergencia. El algoritmo de *Maximización de la Esperanza* (ME) proporciona un marco interactivo para la EMV con aceptables propiedades de convergencia.

El proceso de estimación de parámetros se lleva a cabo dentro de un paradigma de aprendizaje supervisado en el que el reconocedor dispone de un gran número de expresiones de ejemplo, junto con sus transcripciones que típicamente consisten en secuencias de palabras. Hay que tener en cuenta que la información sobre la separación de las señales que son expresiones del habla, de aquellas que no lo son no es necesaria ya que el paradigma de aprendizaje supervisado permite a los modelos estadísticos obtener esta información de forma automática.

Por lo tanto, un sistema de reconocimiento del habla realiza una gran cantidad de auto organización durante el proceso de entrenamiento y tiene la flexibilidad para aprender las diferencias sutiles en los datos de entrenamiento. Para entender el modelo acústico, conviene describir el proceso básico de decodificación basado en MOMs. Por decodificación nos referimos al cálculo del $\arg\max_{\overline{v}}$ en la Ec. 2.1.

El proceso comienza con la postulación de una secuencia de palabras $\overline{v}$. Cada palabra en esta secuencia se convierte en una secuencia de unidades básicas del habla llamadas *fonos*[1] usando un diccionario de pronunciación. La probabilidad de cada fono se calcula con un MOM. La probabilidad de la secuencia $p(\overline{\mathbf{x}} \mid \overline{v})$ postulada se calcula concatenando los modelos de los fonos que componen la secuencia. El modelo de lenguaje calcula la *probabilidad a priori* de la secuencia de palabras y selecciona la secuencia más probable y cada fono se representa con un MOM. Suponiendo que $\mathcal{Q}$ es el conjunto de todos los estados y dado que $\overline{q}$ es una secuencia de estados, es

[1]Mientras que un *fonema* representa la unidad más pequeña de información en el habla, su realización acústica, denotada como *fono* puede ser muy variable en sus propiedades. La pronunciación del fono ofrece una serie de pistas o indicaciones acústicas del habla al oyente, quien debe ser capaz de deducir a partir de ellas el fonema subyacente [Benesty et al., 2008, p. 74].

decir $\overline{q} = (q_1, q_2, ..., q_T)$, se asume que existe alguna variable aleatoria latente $q_t \in \mathcal{Q}$ para cada trama $\mathbf{x}_t$ de $\overline{\mathbf{x}}$. El MOM de cada fono típicamente esta compuesto por tres estados (que representan a las variables ocultas o latentes) en topologías de izquierda a derecha, como se ilustra en la Fig. 2.3.

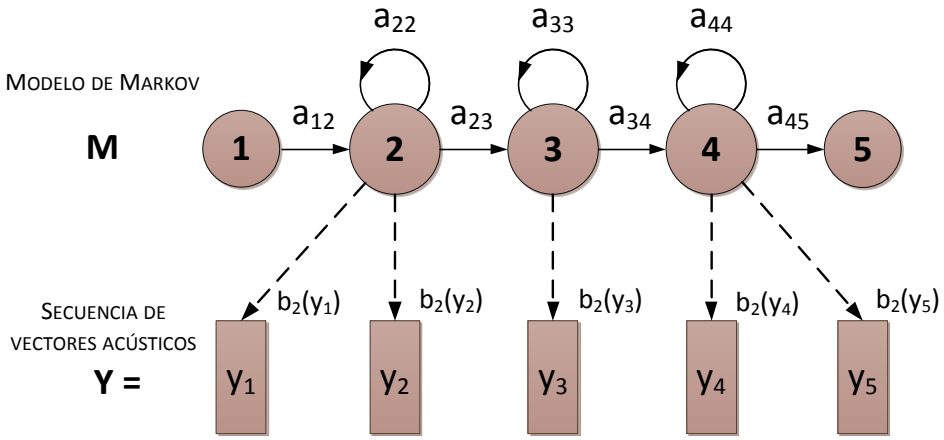

FIGURA 2.3: Topología básica del modelo oculto de Markov para un fono. Cada fono típicamente se representa con un MOM de izquierda a derecha de tres estados. La probabilidad de transición del estado i al estado j estimada se denota como a_{ij}. La probabilidad de salida estimada de un vector acústico $\mathbf{x}_t$, dado que fue emitido desde el estado i, se denota como $b_i(\mathbf{x}_t)$.

En resumen, la secuencia de palabras $\overline{v}$ se convierte en una secuencia de fonos $\overline{p}$ usando un diccionario de pronunciación, y la secuencia de fonos se convierte a una secuencia de estados con por lo general al menos tres estados por fono. El objetivo ahora es encontrar la secuencia de estados más probable. Formalmente se define a un MOM como el par de procesos aleatorios $\overline{q}$ y $\overline{\mathbf{x}}$, en donde se hacen las siguientes suposiciones de Markov de primer orden:

$$P(q_t|q_1, q_2, ..., q_{t-1}) = P(q_t|q_{t-1}) \text{ y}$$

$$p(\mathbf{x}_t|\mathbf{x}_1, ..., \mathbf{x}_{t-1}, \mathbf{x}_{t+1}, ..., \mathbf{x}_T, q_1, ..., q_T) = p(\mathbf{x}_t|q_t).$$

Los MOMs pueden por lo tanto considerarse generadores de secuencias de vectores acústicos es decir se trata de *modelos generativos*. Durante cada

unidad de tiempo o trama, el modelo puede cambiar un estado con probabilidad $P(q_t|q_{t-1})$ también conocido como *probabilidad de transición*. Luego con cada cambio en unidad de tiempo, se emite un vector con probabilidad $p(\mathbf{x}_t|q_t)$, algunas veces referida como *probabilidad de emisión*. En la práctica la secuencia de estados no es observable; y es por eso que al modelo se le llama oculto. La probabilidad de la secuencia de estado $\overline{q}$ dada la secuencia de observación $\overline{\mathbf{x}}$ puede encontrarse usando la regla de Bayes como sigue:

$$P(\overline{q}|\overline{\mathbf{x}}) = \frac{p(\overline{\mathbf{x}}, \overline{q})}{p(\overline{\mathbf{x}})},$$

en donde la probabilidad conjunta de una secuencia de vectores $\overline{\mathbf{x}}$ y una secuencia de estados $\overline{q}$ se calcula simplemente como el producto de las probabilidades de transición y las probabilidades de salida,

$$p(\overline{\mathbf{x}}, \overline{q}) = P(q_0)\prod_{t=1}^{T} P(q_t|q_{t-1})p(\mathbf{x}_t|q_t), \tag{2.3}$$

asumimos que q_0 está restringida a ser un estado inicial no emisor. Las funciones (distribuciones) de densidad de la probabilidad de emisión $p(\mathbf{x}_t|q_t)$ comúnmente se calculan usando la matriz de covarianza diagonal de los Modelos de Mezclas Gaussianas (MMGs) para cada estado q_t, las cuales modelan la densidad de un vector d-dimensional $\mathbf{x}$ como sigue:

$$p(\mathbf{x}) = \sum_i \omega_i \mathcal{N}(\mathbf{x}; \mu_i, \sigma_i), \tag{2.4}$$

en donde $\omega_i \in \mathbb{R}$ es positiva con $\sum_i \omega_i = 1$, y $\mathcal{N}(\cdot; \mu, \sigma)$ es una Gaussiana con media $\mu_i \in \mathbb{R}^d$ y desviación estándar $\sigma_i \in \mathbb{R}^d$. Dados los parámetros del MOM en la forma de probabilidades de transición y de emisión (en forma de MMG), el problema de encontrar la secuencia de estados más probable se resuelve maximizando $p(\overline{\mathbf{x}}, \overline{q})$ sobre todas las secuencias de estado posible usando el *algoritmo Viterbi* [Rabiner y Juang, 1993]. Los parámetros del modelo, se estiman en la fase de entrenamiento. Asumiendo que se tiene acceso a un conjunto de datos de entrenamiento con m ejemplos, los modelos acústico y de lenguaje pueden entrenarse en dos pasos por separado $\mathcal{T}_{entrenamiento} = \{(\overline{\mathbf{x}}_i, \overline{v}_i)\}_{i=1}^{m}$.

Los parámetros del modelo acústico incluyen las probabilidades de transición y de emisión y se estiman por medio del *algoritmo Baum-Welch* [Baum et al., 1970], que es un caso un especial del algoritmo de *Maximización de la Esperanza* (ME) cuando es aplicado a los MOMs [Dempster et al., 1977].

Dicho algoritmo permite calcular iterativamente estas probabilidades de forma muy eficiente. Se eligen los parámetros de los MOMs que maximizan la probabilidad de la secuencia del vector acústico $p(\overline{\mathbf{x}})$. Esta probabilidad se calcula de manera eficiente con el algoritmo Baum-Welch [Rabiner y Juang, 1993], también conocido como *forward-backward*. El algoritmo Baum-Welch converge monótonamente en tiempo polinomial (con respecto al número de estados y a la longitud de las secuencias acústicas) a puntos locales estacionarios de la función de verosimilitud.

Los modelos de lenguaje se usan para calcular la probabilidad de una secuencia de palabras dada $P(\overline{v})$, y con frecuencia se estiman con $n-$gramas, en donde la probabilidad de una secuencia de N palabras $(\overline{v}_1, \overline{v}_2, ..., \overline{v}_N)$ se calcula de la siguiente forma,

$$p(\overline{v}) \approx \prod_t p(v_t | v_{t-1}, v_{t-2}, ..., v_{t-N}), \tag{2.5}$$

en donde cada término de la Ec. 2.5 puede calcularse utilizando un corpus grande de documentos escritos, por medio de un simple conteo de las ocurrencias de cada $n-$grama [Manning y Schütze, 1999]. Se han desarrollado varias estrategias de suavizado y de marcha atrás (algoritmos *backoff*) para el caso en que n es grande y por tanto la mayoría de los $n-$gramas pudieran calcularse mal, incluso usando un corpus de texto muy grande [Katz, 1987].

2.2. Limitaciones y problemas potenciales del enfoque probabilístico

Durante las últimas tres décadas, los MOMs han sido los modelos acústicos predominantes en los sistemas de reconocimiento de voz continua y existen importantes razones para ello, ya que son modelos simples, que además proporcionan una gran flexibilidad para modelar realizaciones acústicas del habla continua. Pero a pesar de que la mayoría de las aproximaciones en

el estado del arte del reconocimiento del habla se basan en el uso de MOMs y Modelos de Mezclas Gaussianas (MMGs) también conocidos como MOMs de Densidad Continua (MOMs-DC), estos tienen varias desventajas y limitaciones, como modelos de señales de voz [Ostendorf et al., 1996] y como algoritmo de aprendizaje. Algunas de estas desventajas se discuten a continuación.

Inherentemente los MOMs son inadecuados para modelar el habla, su principal desventaja en este rubro es su hipótesis estadística. Considerando la probabilidad condicional de la secuencia de vectores acústicos, dada la secuencia de estados, y sustituyendo con $p(\overline{\mathbf{x}}, \overline{q}) = \prod_{t=1}^{T} p(q_t \mid q_{t-1}) p(\mathbf{x}_t \mid q_t)$ en la segunda ecuación tenemos que,

$$P(\overline{\mathbf{x}} \mid \overline{q}) = \frac{p(\overline{\mathbf{x}}, \overline{q})}{p(\overline{q})} = \frac{\prod_{t=1}^{T} p(q_t \mid q_{t-1}) p(\mathbf{x}_t \mid q_t)}{\prod_{t=1}^{T} p(q_t \mid q_{t-1})} = \prod_{t=1}^{T} p(\mathbf{x}_t \mid q_t),$$

Los MOMs suponen independencia condicional de las observaciones dada la secuencia de estados, y si bien esta hipótesis es muy útil, en la práctica resulta físicamente improbable. Otra limitación de los MOM se refiere a la duración inferida de los fonos. El modelo de duración del estado de un MOM está dado implícitamente por una distribución geométrica. La probabilidad de d observaciones consecutivas en el estado i está dada por

$$p(q_{t+1} = j \mid q_t = i)^{d-1} [1 - p(q_{t+1} = j \mid q_t = i)].$$

Debido a que cada fono típicamente se compone de tres estados, la longitud del fono también está dada por una distribución geométrica. Para la mayoría de las señales físicas, una representación de la duración del estado en forma de exponencial descendente no es apropiado [Rabiner y Juang, 1993]. La insuficiencia de los MOMs para modelar la señal de voz es más aguda cuando se compara el modelo de emisión o salida y el modelo de transición. Considerando la Ec. 2.3 en su forma logarímica,

$$\log p(\overline{\mathbf{x}}, \overline{q}) = \log P(q_0) + \sum_{t=1}^{T} \log P(q_t | q_{t-1}) + \sum_{t=1}^{T} \log p(\mathbf{x}_t | q_t) \tag{2.6}$$

El primer término modela las estructuras dinámica y temporal de la señal, mientras que el segundo modela la secuencia de salida. Existe un conocido problema estructural cuando se mezclan densidades $p(\mathbf{x}_t|q_t)$ y probabilidades $P(q_t|q_{t-1})$ ya que la verosimilitud global se encuentra influenciada mayormente por las distribuciones de emisión y casi nada por las probabilidades de transición, por lo tanto los aspectos temporales casí no se toman en cuenta [Bourlard, 1995; Young, 1996].

Esto sucede principalmente debido a que la varianza de las distribuciones de densidad de de la probabilidad de emisión dependen de d, la dimensión real de las características acústicas. Cuanto mayor es d, mayor será la varianza esperada de $p(\overline{\mathbf{x}}, \overline{q})$, en tanto que la varianza de las distribuciones de transición depende principalmente del número de estados del MOM. En la práctica, se puede observar una proporción de alrededor de 100 entre estas varianzas, por lo tanto cuando seleccionamos la mejor secuencia de palabras para una secuencia acústica determinada, sólo las distribuciones de emisión son tomadas en cuenta. Aunque las distribuciones de emisión podrían estimarse mejor usando MMGs, estas no tomarían en cuenta la mayoría de las dependencias temporales entre ellas, que se supone deberían ser modeladas por las transiciones.

Mientras que el algoritmo ME esta bien estudiado y se ha implementado eficientemente para los MOMs, también se sabe que sólo converge a un óptimo local, por lo tanto la optimización puede variar enormemente de acuerdo con los ajustes de los parámetros iniciales. Para los MOMs-DC las medias y varianzas Gaussianas con frecuencia se inicializan usando el algoritmo k-medias, que es conocido por ser de por sí, muy sensible a la inicialización.

El algoritmo de maximización de la esperanza (ME) no sólo es conocido por ser propenso a los óptimos locales, sino que se utiliza básicamente para maximizar la verosimilitud de la secuencia acústica observada en el contexto de la secuencia de palabras esperada. Sin embargo hay que tomar en cuenta que el rendimiento de muchos de los reconocedores del habla se calcula utilizando otras medidas distintas a la verosimilitud. En general, interesa reducir al mínimo el número de errores en la secuencia de palabras generada. Esto se hace con frecuencia mediante el cálculo de la distancia de Levenshtein entre la secuencia de palabras obtenidas y las esperadas, y con frecuencia es conocida como la *tasa de error de palabra*. Así que puede haber una diferencia

significativa entre los MOMs más eficientes de acuerdo con los criterios de máxima verosimilitud y de la tasa de error de palabra.

Por lo anterior a lo largo de los años, se han propuesto varias alternativas. Una línea de investigación se ha centrado en proponer algoritmos discriminatorios de aprendizaje para MOMs. En esta línea se incluyen los algoritmos de *Estimación de Máxima de Información Mutua* [Bahl et al., 1986], de *Error Mínimo de Clasificación* [Juang y Katagiri, 1992], de E*rror Mínimo de Fono* y *Error Mínimo de Palabra* [Povey y Woodland, 2002] por mencionar sólo algunas de las propuestas alternativas más relevantes. Sin embargo aunque todas estas aproximaciones proponen mejores criterios de entrenamiento, aún sufren de la mayor parte de las desventajas descritas anteriormente (mínimos locales y transiciones prácticamente inservibles).

Como algoritmo de aprendizaje, los MOMs tienen también varias desventajas. Hay que recordar que el problema del aprendizaje (estimación) de los parámetros del modelo es un problema de optimización, donde el objetivo es la función de verosimilitud, la cual generalmente no es una función convexa de los parámetros del modelo. De ahí que el algoritmo de Baum-Welch, que calcula los parámetros del modelo, garantiza la convergencia a puntos locales estacionarios de la función de verosimilitud en lugar de a su punto global [Jelinek, 1998].

Otro problema con los MOM es que no abordan directamente las tareas discriminatorias y tampoco minimizan la pérdida específica que tiene origen de forma natural en cada tarea. Como por ejemplo la distancia Levenshtein en la tarea de reconocimiento de secuencias de fonemas (también conocida como distancia de edición), que se utiliza para evaluar el desempeño del reconocedor. Los MOMs no son capaces de minimizar la distancia de Levenshtein entre la secuencia de fonemas pronosticada (basándose en el modelo obtenido) y la que es correcta, ya que estos modelos no hacen más que calcular las probabilidades de la secuencia más plausible. Otro inconveniente de los MOMs es la gran cantidad de parámetros del modelo necesarios para estimarlos y su tendencia a sobre ajustar los datos de entrenamiento.

Como se mencionó, se han propuesto algunas variaciones de MOMs para enfrentar cada uno o partes de estos problemas. Sin embargo y dado que los MOMs no son inherentemente un modelo adecuado para el habla, todos estos

modelos tienen sus propias deficiencias. Por otra parte, la hipótesis fundamental para la conformación de un proceso de Markov, que inherentemente impone que la probabilidad de que el estado actual (una parte del fono) es independiente del pasado, dado el estado anterior, no puede ser válido en el caso de una señal de voz. Cabe señalar que existen muchos otros modelos acústicos que no están basados en MOM [V.gr. Digalakis et al., 1992; Fink et al., 2006], sin embargo estos modelos no son ampliamente utilizados, sino sólo como solución parcial o han sido ineficientes para solucionar el problema del modelado acústico.

En este trabajo nos concentramos en un presentar un nuevo modelo acústico para el habla continua ya que consideramos que un buen modelo acústico es de enorme importancia. A través de los años se han hecho nuevas propuestas en las tres grandes áreas de reconocimiento del habla continua, es decir *la extracción de características*, *el modelado acústico* y *del lenguaje*. Mientras que la extracción de características y el modelado del lenguaje se basan en las técnicas más actualizadas, el modelado acústico esta considerablemente atrás, debido principalmente a ideas equivocadas que consideramos incompatibles con la evidencia empírica. Una de estas equivocaciones es que el mejoramiento del modelado del lenguaje para el reconocimiento de voz continua va a resolver por completo el problema de la robustez en los sistemas de RAH [Allen, 2005, Cap. 1].

Los modelos de lenguaje que se construyen mediante el procesamiento del contexto (por ejemplo cadenas de Markov y las gramáticas formales) no desempeñan un papel importante en el reconocimiento en tanto la tasa de error del modelo acústico sea menor que un número crítico. Se podría discutir acerca de este valor crítico, pero un número razonable debe estar cerca de un error de fono del 50 %. Hasta que este valor crítico es alcanzado, el lenguaje (y por lo tanto, cualquier modelo de lenguaje) no tiene suficiente información de entrada para mejorar significativamente el desempeño. En cambio cuando el desempeño alcanza el 100 %, el lenguaje (el contexto) juega un papel fundamental. Así, el lenguaje en tanto que es muy importante, desempeña un rol sólo cuando el resultado está por encima de este umbral crítico y no desempeña ninguno en la comprensión del problema de robustez. De hecho, cuando no se controla (por ejemplo en altas tasas de error en el modelo acústico), el contexto se interpone en la forma de entender la robustez y se convierte en el problema [Allen, 2005, Cap. 1].

2.3. Máquinas de vectores de soporte

El enfoque de aprendizaje automático basado en métodos núcleo y de margen amplio mejor conocido son las *Máquinas de Vectores de Soporte* (MVS) [Vapnik, 1998]. Si bien es cierto que este enfoque no se ha desarrollado especialmente para el procesamiento señales de voz o el reconocimiento automático del habla, el presente trabajo propone el uso de métodos núcleo que se encuentran de una u otra forma inspirados en las MVS.

Utilizando el principio de inducción de *Minimización del Riesgo Estructural* (MRE) como proceso de inferencia, se desea aprovechar algún método de aprendizaje que permita dar respuesta al *problema general de aprendizaje a partir de ejemplos*. El tipo de problema de aprendizaje para el que resulta mas sencillo encontrar un método, ofreciendo una solución de fácil implementación es la clasificación binaria mediante hiperplanos, mejor conocido como MVS para clasificación.

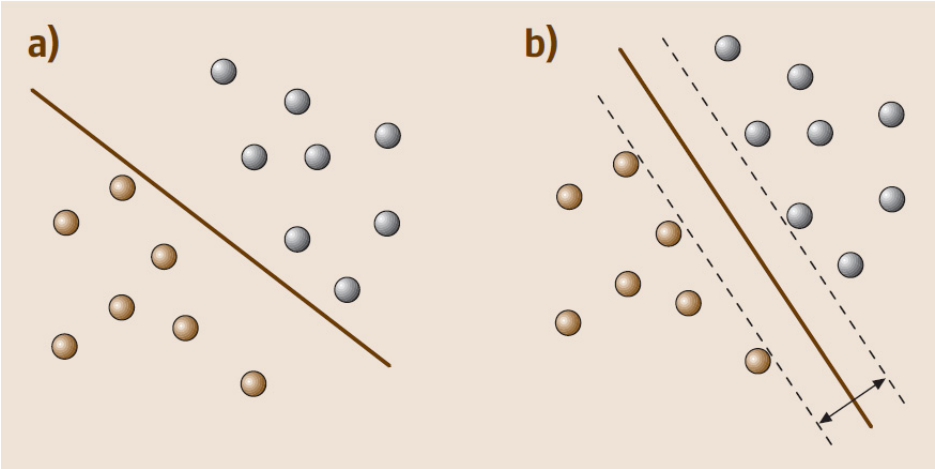

Figura 2.4: Clasificación lineal de margen amplio. **(a)** Elección arbitraria de un hiperplano para separar dos grupos de datos. **(b)** El hiperplano de margen máximo ofrece mayores garantías teóricas.

2.3.1. Máquinas de vectores de soporte para clasificación

Deseamos construir un hiperplano que separe dos clases, de forma que la distancia entre el hiperplano óptimo y el patrón de entrenamiento más cercano sea máxima, con la intención de forzar la generalización de la máquina de aprendizaje [Burges, 1998; Schölkopf y Smola, 2002; Vapnik, 1995]. Para ello comenzamos abordando el problema de clasificación de dos grupos de datos como los que se se muestran en la Fig. 2.4, en donde se desea separar dos poblaciones (esferas obscuras y claras). En este ejemplo simple se puede elegir un hiperplano para separar las dos poblaciones correctamente, pero hay un número infinito de opciones para la selección de ese hiperplano. Existe teoría que permite la elección de un hiperplano que maximiza el *margen*, es decir la distancia que hay entre cada población y el hiperplano de separación. Si decimos, que $\mathcal{F}$ denota la clase de las funciones de valor real en la bola de radio R en $\mathbb{R}^N$, tenemos que

$$\mathcal{F} = \{x \longmapsto w \cdot x : \|w\| \leq 1, \|x\| \leq R\}. \tag{2.7}$$

Entonces, se puede mostrar [Bartlett y Taylor, 1999] que existe una constante c tal que, para todas las distribuciones D sobre X, con probabilidad de por lo menos $1 - \delta$, si un clasificador $\text{sign}(f)$ con $f \in \mathcal{F}$ tiene un margen de por lo menos ρ en m ejemplos de entrenamiento generados independientemente, entonces el error de generalización de $\text{sign}(f)$, es decir el error en cualquier ejemplo futuro, no es mayor a

$$\frac{c}{m}\left(\frac{R^2}{\rho^2}\log^2 m + \log\frac{1}{\delta}\right). \tag{2.8}$$

Este límite y su existencia justifican el uso de los algoritmos de clasificación de margen amplio, tales como las MVS. Sea $w \cdot x + b = 0$ la ecuación del hiperplano, donde $w \in \mathbb{R}^N$ es un vector normal al hiperplano y $b \in \mathbb{R}$ es un valor escalar. El clasificador $\text{sign}(h)$ correspondiente a este hiperplano es único y puede definirse con respecto a los puntos de entrenamiento $x_1, ..., x_m$ de la siguiente forma,

$$h(x) = w \cdot x + b = \sum_{i=1}^{m} \alpha_i (x_i \cdot x) + b, \tag{2.9}$$

en donde α_i son coeficientes de valor real. El punto principal que nos interesa aquí es que, tanto para la construcción de la hipótesis como para el uso posterior de esa hipótesis en la clasificación de nuevos ejemplos, sólo es necesario calcular un cierto número de productos punto entre los ejemplos.

En la práctica, la separación lineal de los datos de entrenamiento no siempre es posible. La Fig. 2.5.a muestra un ejemplo en donde el hiperplano atraviesa ambas poblaciones. Sin embargo, es posible utilizar funciones más complejas para separar los dos grupos como en la Fig. 2.5.b. Una forma de hacerlo es utilizar un mapeo no lineal $\Phi : X \rightarrow F$ del espacio de entrada X a un espacio F de dimension superior[2], en donde donde la separación lineal sea posible.

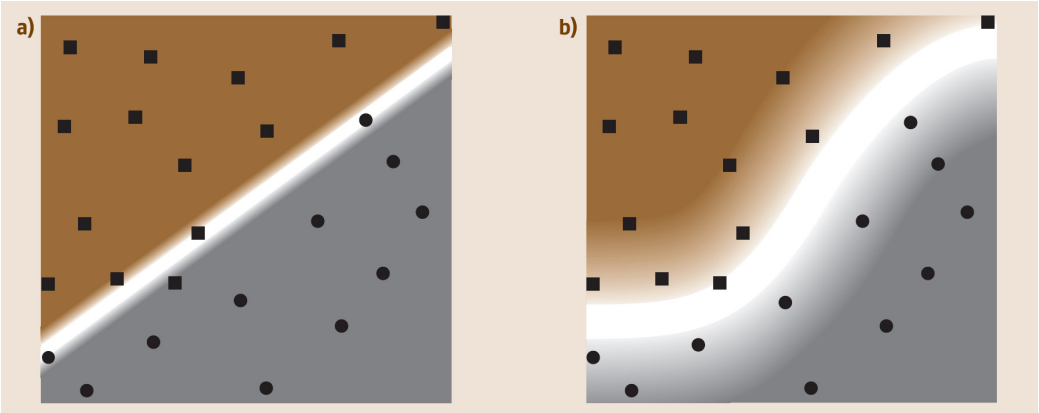

Figura 2.5: El caso no lineal separable de la clasificación de datos. La tarea de clasificación consiste en distinguir entre los cuadrados y los círculos. **(a)** No existe hiperplano que pueda separar las dos poblaciones. **(b)** Un mapeo no lineal se puede utilizar en su lugar.

En la práctica la dimensión de F puede ser realmente muy grande. Por ejemplo, en el caso de la clasificación de documentos se pueden utilizar como características secuencias de tres palabras consecutivas (trigramas). Por lo tanto, con un vocabulario de apenas 100,000 palabras, la dimensión del espacio de características F es de 10^{15}. Pero por otra parte, como lo indica el límite del error en la Ec. 2.8, la capacidad de generalización de los clasificadores de margen amplio como las MVS, no depende de la dimensión del

[2]Se denomina espacio de características y esta dotado de producto interno, vía una inyección no lineal.

espacio de características, sino sólo del margen ρ y el número de ejemplos de entrenamiento m. Sin embargo, realizar un gran número de productos punto en un espacio de gran dimensión para definir el hiperplano puede llegar a ser computacionalmente muy costoso.

Una solución a este problema es usar los llamados métodos núcleo. Una idea central en este enfoque es definir una función $K : X \times X \to \mathbb{R}$ llamada *función núcleo*, de modo que para dos ejemplos x e y en el espacio de entrada $K(x, y)$, la función núcleo sea equivalente al producto escalar de dos ejemplos $\Phi(x)$ y $\Phi(y)$ en el espacio de características

$$K(x, y) = \Phi(x) \cdot \Phi(y) \qquad \forall x, y \in X \tag{2.10}$$

en donde K se describe frecuentemente como una medida de similitud. La ventaja crucial de K es la eficiencia, sólo es necesario definir $\Phi(x)$ y $\Phi(y)$ y realizar el cálculo de $\Phi(x) \cdot \Phi(y)$. Otra ventaja de K es la flexibilidad, ya que puede ser elegida arbitrariamente, siempre y cuando la existencia de Φ este garantizada, lo cual se conoce como la condición de Mercer[3]. Algunos núcleos que cumplen con dicha condición sobre un espacio vectorial, también conocidos como *núcleos Mercer estándar*, son los *núcleos polinomiales* de grado $d \in \mathbb{N}$: $K_d(x, y) = (x \cdot y + 1)^d$. Y los *núcleos Gaussianos*: $K_\sigma(x, y) = \exp\left(\frac{-\|x - y\|^2}{\sigma^2}\right)$, donde $\sigma \in \mathbb{R}_+$.

Una condición equivalente a la condición de Mercer es que la función núcleo K *este positivamente definida y sea simétrica*, es decir, en el caso discreto, la matriz $[K(x_i, y_j)]_{1 \leq i,j \leq n}$ debe ser simétrica y positivamente semidefinida para cualquier elección de n puntos $x_1, ..., x_n$ en X. Dicho de otra manera, la matriz debe ser simétrica y con valores propios no negativos. Por lo tanto, para el problema que nos interesa, la cuestión es cómo definir núcleos que sean simétricos y esten positivamente definidos.

Asumiendo que contamos con un conjunto de entrenamiento de m ejemplos, $\mathcal{T}_{entrenamiento} = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$, en donde $\mathbf{x}_i \in \mathbb{R}^d$ es un vector de entrada

[3]Esta condición es importante para garantizar la convergencia de algoritmos aprendizaje como las MVS.

d-dimensional y $y_i \in \{-1, 1\}$ es la clase de destino (etiquetas de clase). Estamos en búsqueda de los parámetros ($\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}$) del clasificador binario simple (clasificador lineal), tales que

$$\hat{y}(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) \tag{2.11}$$

Entonces, se dice que el conjunto de entrenamiento es linealmente separable cuando potencialmente existe un número infinito de soluciones ($\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}$) que satisfacen la Ec. 2.11. El enfoque basado en máquinas de vectores de soporte (MVS), busca aquella solución que maximiza el margen entre las dos clases, donde dicho margen también puede definirse como la suma de las menores distancias entre la separación del hiperplano y los puntos de cada clase. Esto puede expresarse como el siguiente problema de optimización:

$$\min_{\mathbf{w},b} \frac{1}{2} \|\mathbf{w}\|^2, \qquad \text{sujeto a } \forall i \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1. \tag{2.12}$$

Dado que expresado de esta forma, el problema resulta difícil de resolver, su siguiente formulación dual resulta computacionalmente más eficiente

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j, \qquad \text{sujeto a } \begin{cases} \forall i \quad \alpha_i \geq 0 \\ \sum_{i=1}^{m} \alpha_i y_i = 0 \end{cases} \tag{2.13}$$

Un inconveniente con esta formulación es que si el problema no es linealmente separable, puede no existir una solución para este. Por lo tanto se pueden relajar las restricciones permitiendo errores, con un hiperparámetro adicional C que ajuste el equilibrio entre la maximización del margen y la minimización del número de errores de entrenamiento [Cortes y Vapnik, 1995], de la siguiente manera

$$\min_{\mathbf{w},b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \xi_i, \quad \text{sujeto a } \begin{cases} \forall i \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \\ \forall i \quad \xi_i \geq 0 \end{cases} \tag{2.14}$$

la cual tiene la siguiente formulación dual:

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j, \quad \text{sujeto a} \begin{cases} \forall i \quad 0 \leq \alpha_i \leq C \\ \sum_{i=1}^{m} \alpha_i y_i = 0. \end{cases} \tag{2.15}$$

Con el fin de buscar soluciones no lineales, se puede fácilmente reemplazar $\mathbf{x}$ por alguna función no lineal $\phi(\mathbf{x})$. Es interesante notar que $\mathbf{x}$ sólo aparece en los productos punto en la Ec. 2.15. Así que lo que se ha propuesto es reemplazar todas las ocurrencias de $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ por alguna *función núcleo* $k(\mathbf{x}_i, \mathbf{x}_j)$. Ya que mientras $k(\cdot, \cdot)$ exista en un *Espacio de Hilbert Generado por Núcleo* (EHGN[4]), se puede garantizar que existe alguna función $\phi(\cdot)$ tal que

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j).$$

Por lo tanto incluso si $\phi(\mathbf{x})$ proyecta a $\mathbf{x}$ en un espacio de dimensión muy grande (posiblemente infinita), el núcleo $k(\mathbf{x}_i, \mathbf{x}_j)$ todavía puede ser calculado eficientemente.

El problema descrito en la Ec. 2.15 se puede resolver utilizando herramientas de optimización cuadrática. Como se mencionó antes, hay que tener en cuenta, que la complejidad computacional subyacente es al menos de segundo grado para el número de ejemplos de entrenamiento, lo cual a menudo puede ser un serio límite para la muchas de las aplicaciones de procesamiento del señales de voz. Después de resolver la Ec. 2.15, la MVS resultante toma la siguiente forma

$$\hat{y}(\mathbf{x}) = sign\left(\sum_{i=1}^{m} y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b \right) \tag{2.16}$$

en donde la mayoría de los α_i son cero, excepto aquellos correspondientes a los ejemplos en el margen o clasificados erróneamente, a menudo llamados vectores de soporte (y de ahí el nombre de *máquinas de vectores de soporte*).

[4]Del inglés Reproducing Kernel Hilbert Space (RKHS).

Capítulo 3
Clasificación de fonemas

Resumen:

Se presenta un marco algorítmico para la clasificación de fonemas, en donde se organiza al conjunto de fonemas del idioma inglés en una estructura jerárquica predefinida que se codifica a partir de un árbol, que a su vez induce una medida aplicable al conjunto de fonemas. Éste enfoque combina técnicas de los métodos núcleo, de margen amplio y del análisis Bayesiano. Para extender la noción de margen amplio a dicha clasificación jerárquica, se asocia un prototipo con cada fonema individual y con cada grupo fonético que corresponde a un nodo en la estructura de árbol. Posteriormente formulamos la tarea de aprendizaje como un problema de optimización con restricciones de margen sobre el conjunto de fonemas. De forma similar a los métodos Bayesianos, se imponen requisitos de similitud entre los prototipos correspondientes a los fonemas adyacentes en la jerarquía fonética. Se describe un algoritmo para resolver el problema de clasificación jerárquica. En la parte experimental se demuestra la utilidad del enfoque presentado mediante la aplicación del algoritmo a señales de voz sintética así como en señales de voz natural.

3.1. Introducción

Definimos a la clasificación de fonemas como la tarea de decidir cual es la identidad fonética de una expresión hablada. Es decir consiste en encontrar la mejor secuencia posible de fonemas que describan a una determinada frase pronunciada. Consideremos la situación que se muestra en la Fig. 3.1. Dada una expresión hablada (etiquetada como *(a)* en la figura), que dividimos en pequeños fragmentos del habla, también conocidos como tramas (*(b)* en la figura). Los cuales convertimos en vectores de características acústicas de longitud fija *(c)*, que serán los valores de entrada para la máquina de aprendizaje *(d)*. El objetivo del aprendizaje es clasificar cada uno de estos vectores en uno de los 39 posibles fonemas del idioma Inglés *(e)*, de acuerdo con la estructura jerárquica propuesta. Las predicciones realizadas por la máquina de aprendizaje pueden posteriormente ser la entrada de un modelo estadístico para encontrar la secuencia de fonemas más probable que construye una frase con sentido. Por lo tanto, la clasificación usando tramas puede verse como una de las unidades básicas de un sistema automático de reconocimiento del habla.

El trabajo en el reconocimiento del habla y en particular en la clasificación de fonemas usualmente impone el supuesto de que los diferentes errores de clasificación son de la misma importancia. Sin embargo y dado el conjunto de fonemas descrito por la estructura jerárquica, nos parece necesario considerar que algunos errores tienden a ser más tolerables que otros. Consideramos que muchas veces una predicción para una expresión determinada no se puede declarar como altamente confiable sin previamente ser capaces de identificar con precisión el grupo fonético al que pertenece dicha expresión. En este trabajo se propone y analiza un modelo jerárquico de clasificación que le impone la noción de "gravedad" a los errores de predicción de acuerdo con una estructura jerárquica previamente definida.

La teoría fonética agrupa al conjunto de fonemas de las lenguas occidentales en una jerarquía fonética, en la cual los fonemas constituyen las hojas de los árboles, mientras que los grupos fonéticos grandes, como por ejemplo los correspondientes a las vocales y consonantes, corresponden a los vértices internos. Podemos encontrar descripciones de tales árboles fonéticos en Deller et al. [1999, pp. 115-145] y Rabiner y Schafer [1978, pp. 42-55].

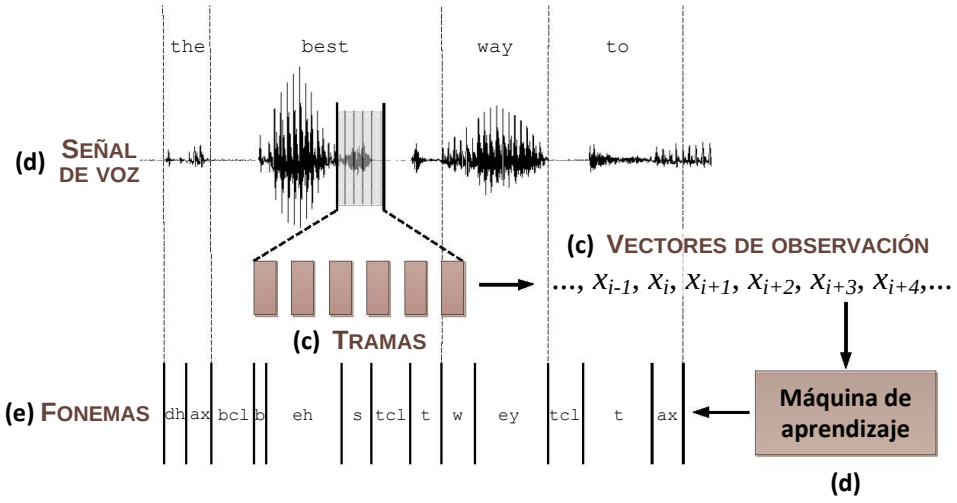

Figura 3.1: Descripción general de la clasificación basada en tramas. **(a)** Señal de voz. **(b)** Tramas con fragmentos de la señal de voz. **(c)** Vector de características acústicas de longitud fija. **(d)** Máquina de aprendizaje que realiza la clasificación del vector de características en uno de 39 posibles fonemas organizados en una estructura jerárquica. **(e)** Fonemas del idioma Inglés.

Una de las primeras propuestas que involucran el reconocimiento de fonemas usando la base de datos TIMIT fue presentada por Lee y Hon [1989], justo después de que TIMIT fuera liberado en diciembre de 1988. Su sistema está basado en MOM discretos[1] y alcanzó una tasa del 73,8 % de exactitud y una del 66,08 % de precisión usando 160 expresiones del habla (frases) de uno de los subconjuntos dispuestos para pruebas (TID7) en el corpus TIMIT. Los autores sugirieron que sus resultados debían convertirse en un referente para el reconocimiento de fonos con la base de datos TIMIT, y de hecho su trabajo efectivamente se ha convertido en un referente, pero no sólo por el desempeño alcanzado sino también por el plegamiento de fonemas que ahí se propone[2]. Los autores plegaron las 61 etiquetas para fonos usadas en la base

[1]Sus mejores resultados se lograron con fonos modelados mediante 1450 difonos, utilizando un modelo de lenguaje de bigramas y tres libros de códigos (codebooks) con 256 vectores prototipo de coeficientes cepstrales de predicción lineal que fueron utilizados como secuencias de características acústicas.

[2]La descripción detallada el conjunto de original de fonos que usa el corpus TIMIT se

de datos TIMIT en 48 fonemas para fines de entrenamiento, y para efectos de evaluación plegaron las 61 etiquetas TIMIT en 39 fonemas, lo cual se ha convertido en el estándar para evaluación de sistemas de clasificación de fonemas. En la Tabla 3.1 se describe el proceso de plegamiento y una parte de los 39 fonemas resultantes. Los fonemas en la columna de la izquierda literalmente se doblan dejando como resultado las etiquetas de la columna a la derecha. Desaparecen 23 etiquetas de fonemas y se añade al conjunto la etiqueta de silencio "SIL". El resto de los fonemas del conjunto original de 61 permanecen intactos.

aa, ao	aa
ah, ax, ax-h	ah
er, axr	er
hh, hv	hh
ih, ix	ih
l, el	l
m, em	m
n, en, nx	n
ng, eng	ng
sh, zh	sh
uw, ux	uw
pcl, tcl, kcl, bcl, dcl, gcl, h#, pau, epi	*sil*
q	-

TABLA 3.1: Mapeo de las 61 clases de fonemas originales del corpus TIMIT a 39 clases, según lo propuesto por Lee y Hon [1989]. Los fonemas en la columna de la izquierda se pliegan para formar las etiquetas en la columna de la derecha. Los fonemas restantes permanecen intactos. El fonema "q" se descarta.

A pesar de los avances logrados en las últimas décadas, la tarea de reconocimiento de fonemas usando el corpus TIMIT sigue siendo una tarea compleja. Se han hecho muchos intentos para mejorar el rendimiento de los reconocedores de fonemas, incluyendo el uso de mejores secuencias de características acústicas o de varios conjuntos de dichas secuencias, el mejoramiento

pueden revisar en el *Apéndice A* del presente trabajo.

de los modelos estadísticos, el uso de distintos criterios de entrenamiento, modelado de la pronunciación y distintos modelos acústicos, el manejo del ruido y el modelado del lenguaje, entre otros.

Hacer una descripción de la investigación que en materia de reconocimiento de fonemas usando la base de datos TIMIT se ha realizado en los últimos años, tratando de cubrir la amplia gama de tecnologías que se han propuesto está fuera del alcance del presente trabajo. Sin embargo es pertinente mencionar que la elección de dicho corpus estándar para fines de evaluación del método discriminatorio propuesto nos asegura la posibilidad de comparar nuestros resultados con aquellos que en el área del reconocimiento de fonemas se han venido documentado a lo largo de los años y adicionalmente en cada una de las tareas que dicho reconocimiento involucra (por ejemplo la clasificación y el alineamiento de fonemas).

Tomando en cuenta lo anterior y basándonos en estas estructuras fonéticas haremos uso del modelo jerárquico que se ilustra en la Fig. 3.2, el cual se ajusta a los estándares de evaluación descritos y además ya incorpora la noción de similitud y análogamente de disimilitud, entre fonemas y entre grupos fonéticos. Empleamos esta noción en el proceso de aprendizaje que se describe y analiza a continuación.

La mayor parte del trabajo anterior en la clasificación de fonemas deja de lado la estructura jerárquica fonética (véase por ejemplo Clarkson y Moreno [1999]). En el trabajo de Salomon [2001] se utiliza un algoritmo de agrupamiento jerárquico para la clasificación de fonemas, sin embargo el algoritmo que ahí se propone genera un árbol binario que se usa para la construcción de un clasificador de fonemas que a su vez emplea múltiples máquinas vectores de soporte (MVS) para clasificación binaria. Dicha construcción parece haber sido diseñadada por razones de eficiencia más que para capturar la estructura fonética jerárquica.

En el área del aprendizaje automático el problema de la clasificación jerárquica y en particular la clasificación jerárquica de documentos, se ha abordado en numerosas investigaciones (véanse por ejemplo los trabajos de Koller y Sahami [1997]; Mayne y Perry [2009]). Una gran parte del trabajo previo en clasificación jerárquica separa el problema en sub problemas de clasificación independientes mediante la asignación y entrenamiento de un

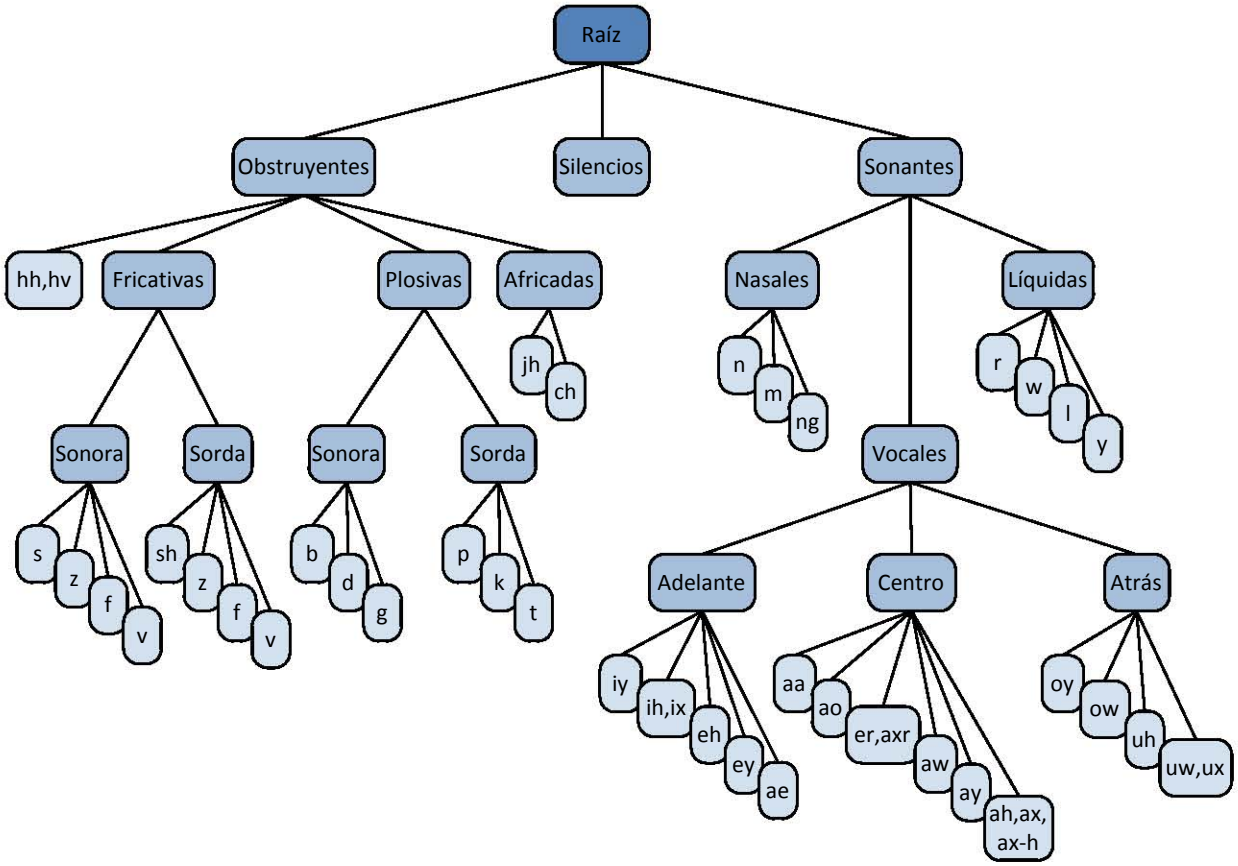

Figura 3.2: Árbol fonético con 39 fonos del idioma Inglés Americano.

clasificador para cada vértice interno dentro de la jerarquía. Con el fin de incorporar la semántica representada en la estructura jerárquica, algunas investigaciones han propuesto imponer restricciones estadísticas de similitud entre los modelos probabilísticos para vértices adyacentes en la jerarquía. En la configuración de las probabilidades, las similitudes estadísticas pueden imponerse usando técnicas tales como estimaciones *hacía atrás* [Katz, 1987] y *de contracción* o *plegamiento* [McCallum et al., 1998].

Recientemente una importante cantidad de trabajos sobre clasificación binaria y multiclase se ha dedicado a establecer la teoría y la aplicación de los clasificadores de margen amplio (véase por ejemplo [Vapnik, 1998]). En este capítulo se describe, analiza y aplica un enfoque de margen amplio a la clasificación jerárquica de fonemas en el marco de los métodos estadísticos. Al igual que en los métodos de margen amplio, se asocia un vector en un espacio de gran dimension con cada fonema o grupo de fonemas en la jerarquía. Llamamos a este vector el *prototipo* del fonema o del grupo de fonemas. Este vector permite a su vez clasificar a los vectores de características de acuerdo a su similitud con los diferentes prototipos. Relajamos los requisitos para la *clasificación correcta* con restricciones de margen amplio y tratamos de encontrar prototipos que cumplan con estas restricciones. En el marco de los métodos Bayesianos, imponemos requisitos de similitud entre prototipos correspondientes a fonemas adyacentes en la estructura jerárquica.

El resultado es un algoritmo capaz de tolerar errores menores, tales como la predicción de un fonema similar (en el mismo nivel de la estructura o hermano) en lugar del que es correcto, pero que simultáneamente elimina errores graves tales como la predicción de un vértice en una parte totalmente diferente del árbol. Para enfrentar el hecho de que los corpus del habla normalmente contienen un gran número de ejemplos (una gran cantidad de datos) se propone un algoritmo que usa eficientemente la memoria y a la vez es fácil de implementar. La solución algorítmica que se presenta se basa en la generalización y fusión de las propuestas hechas en Kivinen y Warmuth [1997], Crammer et al. [2006], Herbster [2001] y Kivinen et al. [2004]. Estos trabajos analizan la tarea de aprendizaje en línea con clasificadores de margen amplio.

En cada ronda, la hipótesis se actualiza de tal manera que cumpla con las restricciones de margen impuestas por el ejemplo observado en esa ronda.

Junto con las restricciones de margen, es necesaria una actualización para mantener el nuevo clasificador lo bastante cerca del anterior. Se muestra que esta idea también puede ser explotada el contexto del problema que intentamos resolver y en la configuración de su solución, lo que resulta en una sencilla actualización en línea que puede ser usada conjuntamente con las funciones núcleo. Además, utilizando métodos para convertir el aprendizaje en línea a aprendizaje por lotes como en el trabajo de Cesa-Bianchi et al. [2004], se muestra que el algoritmo en línea se puede utilizar como base para construir una versión del mismo algoritmo en modo de procesamiento por lotes.

Este capítulo se organiza de la siguiente manera. En la *Sección* 2 se describe formalmente el problema de la clasificación jerárquica de fonemas y se establece la notación que usaremos. En la *Sección* 3 se describe un algoritmo para la clasificación jerárquica de fonemas y se prueba su desempeño para el "peor de los casos", esta sección constituye la parte central del capítulo, adicionalmente en éste se describe en breve la conversión del algoritmo en línea (*online*) en el algoritmo de procesamiento por lotes (*batch*). Y en la *Sección* 4 concluimos este capítulo presentando los resultados de los experimentos realizados con datos de voz sintética y de voz natural que muestran la utilidad del método propuesto.

3.2. Definición del problema

Sea $\mathcal{X} \subseteq \mathbb{R}^n$ el dominio de las secuencias de características acústicas y sea $\mathcal{Y}$ el conjunto de fonemas y grupos de fonemas. En la configuración de la clasificación jerárquica $\mathcal{Y}$ juega un doble papel. En primer lugar, como en los problemas multiclase tradicionales, comprende el conjunto de fonemas, es decir cada vector de características en $\mathcal{X}$ está asociada con un fonema $v \in \mathcal{Y}$. En segundo lugar $\mathcal{Y}$ define un conjunto de vértices, es decir, los fonemas y los grupos de fonemas, organizados en un árbol $\mathcal{T}$. Decimos que $k = |\mathcal{Y}|$ y para ser más concretos asumimos que $\mathcal{Y} = \{0, ..., k-1\}$ donde 0 es la raíz de $\mathcal{T}$. Para cualquier par de fonemas $u, v \in \mathcal{Y}$, sea $\gamma(u, v)$ su distancia en el árbol. Es decir $\gamma(u, v)$ se define como el número de aristas o bordes a lo largo del único camino o ruta que va de u a v en $\mathcal{T}$.

La función de la distancia $\gamma(\cdot,\cdot)$ es una medida sobre $\mathcal{Y}$. Y dado que es una función no negativa, entonces $\gamma(v,v) = 0$, $\gamma(u,v) = \gamma(v,u)$ y además la desigualdad del triángulo se cumple siempre, en dicha igualdad. Como se estableció anteriormente, diferentes errores de clasificación inducen diferentes niveles de penalización y en el modelo sugerido esta penalización se define por la distancia $\gamma(u,v)$ en el árbol. Por lo tanto decimos que el *error inducido por el árbol* en que se incurre dada la predicción del fonema o del grupo de fonemas v cuando el fonema correcto es u será igual a $\gamma(u,v)$.

Contamos con un conjunto de entrenamiento $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$ compuesto por los vectores de características asociados a un fonema, es decir *pares vector-fonema* en donde cada $\mathbf{x}_i \in \mathcal{X}$ y cada $y_i \in \mathcal{Y}$. Nuestro objetivo es aprender una función de clasificación $f : \mathcal{X} \to \mathcal{Y}$ que logre *errores inducidos por el árbol* pequeños. Para ello debemos concentramos en los clasificadores que cumplan con lo siguiente: cada fonema $v \in \mathcal{Y}$ deberá tener un prototipo coincidente $\mathbf{W}^v \in \mathbb{R}^n$, en donde $\mathbf{W}^0$ se configura como el vector cero o nulo y cualquier otro prototipo puede ser cualquier vector en $\mathbb{R}^n$. Y el clasificador f deberá realizar sus predicciones de acuerdo con la siguiente regla

$$f(\mathbf{x}) = \underset{v \in \mathcal{Y}}{\operatorname{argmax}}\, \mathbf{W}^v \cdot \mathbf{x} \tag{3.1}$$

Entonces la tarea de aprendizaje f se reduce a aprender $\mathbf{W}^1, ..., \mathbf{W}^{k-1}$. Para cada fonema o grupo de fonemas que no sea la raíz del árbol $v \in \{\mathcal{Y} \backslash 0\}$, denotamos como $\mathcal{A}(v)$ al padre de v en el árbol, dicho de otra manera $\mathcal{A}(v)$ es el vértice adyacente a v que está más cerca de la raíz del árbol. También definimos $\mathcal{A}^{(i)}(v)$ como el $i-$ésimo antecesor de v (si es que existe uno). Formalmente $\mathcal{A}^{(i)}(v)$ se define recursivamente como $\mathcal{A}^{(0)}(v) = v$, y además, $\mathcal{A}^{(i)}(v) = \mathcal{A}\left(\mathcal{A}^{(i-1)}(v)\right)$.

Para cada fonema o grupo de fonemas $v \in \mathcal{Y}$ definimos a $\mathcal{P}(v)$ como el conjunto de los grupos de fonemas en el camino que va de 0 (la raíz del árbol) a v, es decir:

$$\mathcal{P}(v) = \left\{u \in \mathcal{Y} : \exists i u = \mathcal{A}^{(i)}(v)\right\} .$$

Por razones técnicas preferimos no tratar directamente con el conjunto de prototipos $\mathbf{W}^1, ..., \mathbf{W}^{k-1}$, sino más bien con la diferencia entre cada prototipo

y el prototipo de su padre. Formalmente, se define a $\mathbf{w}^0$ como el vector cero en $\mathbb{R}^n$ y para cada fonema o grupo de fonemas $v \in \{\mathcal{Y} \setminus 0\}$ sea $\mathbf{w}^v = \mathbf{W}^v, ..., \mathbf{W}^{\mathcal{A}(v)}$. Cada prototipo ahora se descompone en siguiente suma:

$$\mathbf{W}^v = \sum_{u \in \mathcal{P}(v)} \mathbf{w}^u. \tag{3.2}$$

Entonces el clasificador f se puede definir de dos formas equivalentes, la primera forma es estableciendo $\{\mathbf{W}^v\}_{v \in \mathcal{Y}}$ y usando la Ec. 3.1, y la segunda es estableciendo $\{\mathbf{w}^v\}_{v \in \mathcal{Y}}$ y usando la Ec. 3.2 junto con la Ec. 3.1. A lo largo de este trabajo, se utiliza a menudo $\{\mathbf{w}^v\}_{v \in \mathcal{Y}}$ como sinónimo de la función f de clasificación. Como elección de diseño, los algoritmos que se presentan requieren que los vértices adyacentes en el árbol fonético tengan prototipos similares. El beneficio de representar cada prototipo $\{\mathbf{W}^v\}_{v \in \mathcal{Y}}$ como una suma de vectores de $\{\mathbf{w}^v\}_{v \in \mathcal{Y}}$ es que los prototipos adyacentes $\mathbf{W}^v$ y $\mathbf{W}^{\mathcal{A}(v)}$ pueden mantenerse cerca simplemente manteniendo a $\mathbf{w}^v = \mathbf{W}^v - \mathbf{W}^{\mathcal{A}(v)}$ pequeño. La *Sección 3* aborda la tarea de aprendizaje del conjunto $\{\mathbf{w}^v\}_{v \in \mathcal{Y}}$ a partir de datos supervisados.

3.3. Algoritmo para clasificación de fonemas

En esta sección se deduce y analiza un algoritmo eficiente de aprendizaje en línea para el problema de la clasificación jerárquica de fonemas. En la ronda i un vector de características acústicas, denotado como $\mathbf{x}_i$, se da como entrada al algoritmo de aprendizaje. El algoritmo mantiene un conjunto de prototipos que se actualizan constantemente de acuerdo con la calidad de sus predicciones. Denotamos al conjunto de prototipos usados para ampliar la predicción en la ronda i como $\{\mathbf{w}_i^v\}_{v \in \mathcal{Y}}$. Por lo tanto decimos que el fonema o grupo de fonemas predichos por el algoritmo para x_i es,

$$\hat{y}_i = \operatorname*{argmax}_{v \in \mathcal{Y}} \mathbf{W}_i^v \cdot \mathbf{x}_i = \operatorname*{argmax}_{v \in \mathcal{Y}} \sum_{u \in \mathcal{P}(v)} \mathbf{w}_i^u \cdot \mathbf{x}_i \ .$$

Entonces cuando se descubre el fonema correcto y_i, en el algoritmo ocurre un error instantáneo. Como se menciona más arriba, el nombre que empleamos para tal error es *error inducido por el árbol*. Utilizando la notación descrita anteriormente el error en la ronda i es igual a $\gamma(y_i, \hat{y}_i)$.

De nuestro análisis así como de la motivación para la actualización en línea que se deduce a continuación, se asume que existe un conjunto de prototipos $\{\omega^v\}_{v\in\mathcal{Y}}$ tal que para cada par *vector-fonema* de características acústicas $(\mathbf{x}_i, y_i)$ y cada $r \neq y_i$ se cumple que

$$\sum_{v\in\mathcal{P}(y_i)} \omega^v \cdot \mathbf{x}_i - \sum_{u\in\mathcal{P}(r)} \omega^u \cdot \mathbf{x}_i \geq \sqrt{\gamma(y_i, r)}. \tag{3.3}$$

La diferencia que describe la Ec. 3.3 entre la proyección sobre el prototipo correspondiente al fonema correcto y cualquier otro prototipo es una generalización de la noción de margen empleada para problemas multiclase en la literatura del aprendizaje automático (véase [Weston y Watkins, 1999]). Dicho de otra manera, es necesario que el margen entre el fonema correcto y cada uno de los fonemas y grupos de fonemas incorrectos sea de por lo menos la raíz cuadrada de la distancia entre ellos (distancia basada en el árbol). El objetivo del algoritmo es encontrar un conjunto de prototipos que cumplan con el requisito de margen de la Ec. 3.3 ya que mientras que dicho conjunto no se encuentre se incurre en un error mínimo inducido por el árbol. Sin embargo, dicho error es una cantidad combinatoria y es por lo tanto difícil de reducir directamente. En lugar de intentarlo de esa forma, utilizamos una construcción de uso común en los clasificadores de margen amplio, *la función de pérdida en forma de bisagra convexa* (conocida también como función *hinge-loss*) que tiene la forma

$$\ell\left(\{\mathbf{w}_i^v\}, \mathbf{x}_i, y_i\right) = \left[\sum_{v\in\mathcal{P}(\hat{y}_i)} \mathbf{w}_i^v \cdot \mathbf{x}_i - \sum_{v\in\mathcal{P}(y_i)} \mathbf{w}_i^v \cdot \mathbf{x}_i + \sqrt{\gamma(y_i, \hat{y}_i)}\right]_+, \tag{3.4}$$

en donde $[z]_+ = \text{máx}\{z, 0\}$. Se puede demostrar que $\ell^2\left(\{\mathbf{w}_i^v\}, \mathbf{x}_i, y_i\right)$ es el límite superior de $\gamma(y_i, \hat{y}_i)$ y utilizamos este hecho para obtener una cota para $\sum_{i=1}^m \gamma(y_i, \hat{y}_i)$. El algoritmo en línea pertenece a la familia de algoritmos llamados *conservadores*, ya que actualizan sus reglas de clasificación sólo en las rondas en las que se cometieron los errores de predicción.

Supongamos que hubo un error de predicción en la ronda i. Queremos modificar el conjunto de vectores $\{\mathbf{w}_i^v\}$ con el fin de satisfacer las restricciones

de margen impuestas para el $i-$ésimo ejemplo. Una posible aproximación es simplemente encontrar un conjunto de vectores que resuelva las restricciones en la Ec. 3.3. Tal conjunto debe existir dado que asumimos que existe un conjunto $\{\omega_i^v\}$ que satisface los requisitos de margen para *todos* los ejemplos. Sin embargo, hay al menos dos consideraciones para un enfoque tan ambicioso. La primera es que cuando establecemos el nuevo conjunto de prototipos como una solución (arbitraria) a las restricciones impuestas por el ejemplo más reciente, nos encontramos en peligro de olvidar lo que se ha aprendido hasta ahora y la segunda es que no hay una solución analítica simple a la Ec. 3.3.

Por lo tanto proponemos tratarlo como un problema simple de optimización con restricciones. La función objetivo de este problema de optimización asegura que el nuevo conjunto $\{\mathbf{w}_{i+1}^v\}$ se mantendrá cerca del conjunto actual, mientras que las restricciones aseguran que el requisito del margen para el par $\gamma(y_i, \hat{y}_i)$ se cumpla para los nuevos vectores. Por ello podemos decir formalmente que el nuevo conjunto de vectores será la solución al siguiente problema,

$$\min_{\{\mathbf{w}^v\}} \frac{1}{2} \sum_{v \in \mathcal{Y}} \left\| \mathbf{w}^v - \mathbf{w}_i^v \right\|^2, \tag{3.5}$$

$$\text{tal que } \sum_{v \in \mathcal{P}(y_i)} \mathbf{w}^v \cdot \mathbf{x}_i - \sum_{u \in \mathcal{P}(\hat{y}_i)} \mathbf{w}^u \cdot \mathbf{x}_i \geq \sqrt{\gamma(y_i, \hat{y}_i)}.$$

En primer lugar, se debe tener en cuenta que cualquier vector $\mathbf{w}^v$ correspondiente a un vértice v que no pertenece ni a $\mathcal{P}(y_i)$ ni a $\mathcal{P}(\hat{y}_i)$ no cambia debido a la función objetivo de la Ec. 3.5, por lo tanto $\mathbf{w}_{i+1}^v = \mathbf{w}_i^v$. En segundo lugar nótese que si $v \in \mathcal{P}(y_i) \cap \mathcal{P}(\hat{y}_i)$ entonces la contribución de $\mathbf{w}^v$ se anula. Por lo tanto, también para este caso tenemos que $\mathbf{w}_{i+1}^v = \mathbf{w}_i^v$. En resumen, los vectores que tenemos que actualizar en realidad corresponden a los vértices en el conjunto $\mathcal{P}(y_i) \Delta \mathcal{P}(\hat{y}_i)$ en donde Δ se usa para denotar la diferencia simétrica de los conjuntos.

Para encontrar la solución a la Ec. 3.5 usamos el multiplicador de Lagrange α_i y reformulamos el problema de optimización en la forma de un

Lagrangiano. Igualando a cero la derivada del Lagrangiano con respecto a $\{\mathbf{w}^v\}$ tenemos que

$$\mathbf{w}_{i+1}^v = \mathbf{w}_i^v + \alpha_i \mathbf{x}_i \; v \in \mathcal{P}(y_i) \setminus \mathcal{P}(\hat{y}_i) \tag{3.6}$$

$$\mathbf{w}_{i+1}^v = \mathbf{w}_i^v - \alpha_i \mathbf{x}_i \; v \in \mathcal{P}(\hat{y}_i) \setminus \mathcal{P}(y_i). \tag{3.7}$$

Dado que en el óptimo la restricción de la Ec. 3.5 es obligatoria, tenemos que

$$\sum_{v \in \mathcal{P}(y_i)} (\mathbf{w}_i^v + \alpha_i \mathbf{x}_i) \cdot \mathbf{x}_i = \sum_{v \in \mathcal{P}(\hat{y}_i)} (\mathbf{w}_i^v + \alpha_i \mathbf{x}_i) \cdot \mathbf{x}_i + \sqrt{\gamma(y_i, \hat{y}_i)}.$$

Reordenando los términos de la ecuación anterior y utilizando la definición de pérdida de la Ec. 3.4 obtenemos

$$\alpha_i \left\|\mathbf{x}_i\right\|^2 \left|\mathcal{P}(y_i) \Delta \mathcal{P}(\hat{y}_i)\right| = \ell\left(\{\mathbf{w}_i^v\}, x_i, y_i\right).$$

Finalmente, dado que la cardinalidad de $\mathcal{P}(y_i)\Delta\mathcal{P}(\hat{y}_i)$ es igual a $\gamma(y_i, \hat{y}_i)$, obtenemos

$$\alpha_i = \frac{\ell\left(\{\mathbf{w}_i^v\}, \mathbf{x}_i, y_i\right)}{\gamma(y_i, \hat{y}_i) \left\|\mathbf{x}_i\right\|^2} \tag{3.8}$$

El pseudocódigo del algoritmo en línea o *Algoritmo 1* para clasificación de fonemas se muestra más adelante. Para concluir la parte algorítmica del capítulo, se observa que diferentes núcleos de Mercer pueden fácilmente ser incorporados en el algoritmo. Primero reescribimos la actualización como $\mathbf{w}_{i+1}^v = \mathbf{w}_i^v + \alpha_i^v \mathbf{x}_i$ en donde

$$\alpha_i = \begin{cases} \alpha_i & v \in \mathcal{P}(y_i) \setminus \mathcal{P}(\hat{y}_i) \\ -\alpha_i & v \in \mathcal{P}(\hat{y}_i) \setminus \mathcal{P}(y_i) \\ 0 & 0 \;\text{ en otro caso} \end{cases}$$

Usando esta notación, el clasificador jerárquico resultante se puede también reescribir como

$$f(x) = \operatorname*{argmax}_{v \in \mathcal{Y}} \sum_{u \in \mathcal{P}(v)} \mathbf{w}_i^u \cdot \mathbf{x}_i \tag{3.9}$$

$$= \operatorname*{argmax}_{v \in \mathcal{Y}} \sum_{u \in \mathcal{P}(v)} \sum_{i=1}^{m} \alpha_i^u \mathbf{x}_i \cdot \mathbf{x}. \tag{3.10}$$

Podemos reemplazar los productos internos en la Ec. 3.10 con un operador núcleo general $K(\cdot, \cdot)$ que satisfaga las condiciones de Mercer [Vapnik, 1998]. Pero todavía queda por mostrar que α_i^v se puede calcular basándonos en operaciones núcleo siempre que $\alpha_i^v \neq 0$. Lo anterior es verificable tomando en cuenta que α_i se puede reescribir a partir de la Ec. 3.8 como

$$\alpha_i = \frac{\left[\sum_{v \in \mathcal{P}(\hat{y}_i)} \sum_{j<i} \alpha_j^v K\left(\mathbf{x}_j, \mathbf{x}_i\right) - \sum_{v \in \mathcal{P}(y_i)} \sum_{j<i} \alpha_j^v K\left(\mathbf{x}_j, \mathbf{x}_i\right) + \gamma(y_i, \hat{y}_i) \right]_+}{\gamma(y_i, \hat{y}_i) K\left(\mathbf{x}_i, \mathbf{x}_i\right)}. \tag{3.11}$$

3.3.1. Conversión del algoritmo de clasificación

En la sección anterior se presentó un algoritmo en línea para la clasificación jerárquica de fonemas. Sin embargo como en nuestro caso, a menudo se cuenta de antemano con todo el conjunto de entrenamiento $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ para ser utilizado por el algoritmo de aprendizaje. En ese caso la utilización de un algoritmo de *procesamiento por lotes* (*batch*) resulta más natural. Como antes, el rendimiento del clasificador f para un ejemplo dado $(\mathbf{x}, y)$ se evalúa con respecto al *error inducido por el árbol* $\gamma(y, f(\mathbf{x}))$. En contraste con el aprendizaje en línea (*online*) donde no se hacen suposiciones sobre la distribución de los ejemplos, en la configuración de procesamiento por lotes se asume que los ejemplos están muestreados independientemente, a partir de una distribución $\mathcal{D}$ sobre $\mathcal{X} \times \mathcal{Y}$. Por lo tanto, nuestro objetivo es utilizar

Algoritmo 1 Algoritmo en línea para clasificación de fonemas

Entrada: Conjunto de datos de entrenamiento $S = \{(\mathbf{x}_i, p_i)\}_{i=1}^{m}$.
Árbol de fonemas $\mathcal{T}$.

Inicializar: $\forall v \in \mathcal{P} : \mathbf{w}_v^1 = 0$

Para $i = 1, 2, ..., m$ **hacer**
Recibir el vector de características acústicas $\mathbf{x}_i$.

Predecir el fonema o grupo de fonemas tal que $\hat{y}_i = \underset{v \in \mathcal{Y}}{\operatorname{argmax}} \sum_{u \in \mathcal{P}(v)} \mathbf{w}_i^u \cdot \mathbf{x}_i$.

Recibir el fonema correcto y_i.

Analizar la pérdida $\ell(\{\mathbf{w}_i^v\}, \mathbf{x}_i, y_i)$. [véase Ec. 3.4]

Acualizar
$\mathbf{w}_{i+1}^v = \mathbf{w}_i^v + \alpha_i \mathbf{x}_i \quad v \in T(p_i) \setminus T(p\prime_i)$
$\mathbf{w}_{i+1}^v = \mathbf{w}_i^v - \alpha_i \mathbf{x}_i \quad v \in T(p\prime_i) \setminus T(p_i)$

fin Para

S para obtener un clasificador jerárquico f que logre el mínimo *error esperado inducido por el árbol* $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\gamma(y, f(\mathbf{x}))]$, donde la esperanza se toma a partir de la selección aleatoria de ejemplos provenientes de $\mathcal{D}$.

Tal vez la idea más sencilla sea utilizar el algoritmo en línea descrito en la *Sección 3* como si fuera un algoritmo de procesamiento por lotes, es decir aplicarlo al conjunto de entrenamiento S en algún orden arbitrario y definir a la f obtenida por este proceso como el clasificador. Para fines experimentales a dicha configuración la llamaremos modo de procesamiento por lotes *simple* (*batch simple*). El clasificador resultante es aquel que está definido por el conjunto de vectores $\left\{\mathbf{w}_{m+1}^v\right\}_{v\in\mathcal{Y}}$. En la práctica, esta idea funcionó razonablemente bien, como lo demuestran los experimentos presentados en la siguiente sección. Para ver los resultados experimentales del algoritmo de procesamiento por lotes en sus dos distintas configuraciones, véanse las Tablas 3.4, 3.5 y 3.6. Sin embargo, con una ligera modificación en la configuración del *procesamiento por lotes simple* se obtiene un clasificador significativamente mejor (al que para fines experimentales llamamos solamente *procesamiento por lotes*, con un límite asociado al error esperado inducido por el árbol.

La modificación consiste en definir $\mathbf{w}^v = \frac{1}{m+1}\sum_{i=1}^{m+1}\mathbf{w}_i^v$, para todo $v \in \mathcal{Y}$ en donde $\{\mathbf{w}_i^v | 1 \leq i \leq m+1, v \in \mathcal{Y}\}$ como el conjunto de vectores generados por el algoritmo en línea cuando se aplica a S, y a f como el clasificador definido por $\{\mathbf{w}^v\}_{v\in\mathcal{Y}}$. Es decir, definimos al prototipo para el fonema o grupo de fonemas v que será el promedio de todos los prototipos generados por el algoritmo en línea para v (véase [Cesa-Bianchi et al., 2004]).

3.4. Experimentos

Comenzamos esta sección con una comparación entre los algoritmos en línea y de procesamiento por lotes con clasificadores multiclase estándar que son ajenos a la estructura jerárquica del conjunto de fonemas descrita. Llevamos a cabo experimentos con un conjunto de *datos sintéticos* y con un conjunto de datos de fonemas extraídos de expresiones del *habla natural continua*. Los datos sintéticos se generaron de la siguiente manera: se construyó un árbol simétrico ternario de profundidad igual a 4 y se utilizó como estructura jerárquica. Éste árbol contiene 121 vértices que son

los "fonemas" de nuestro problema multiclase. A continuación, establecimos $\mathbf{w}^0, \mathbf{w}^1, ..., \mathbf{w}^{119}, \mathbf{w}^{120}$ como un conjunto ortonormal en $\mathbb{R}^{121}$ y definimos 121 prototipos como $\mathbf{W}^v = \sum_{u \in \mathcal{P}(v)} \mathbf{w}^u$. Generamos 100 vectores de entrenamiento y 50 vectores de prueba para cada "fonema" en el árbol con estructura jerárquica. Cada ejemplo se generó estableciendo $(\mathbf{x}, y) = (\mathbf{W}^y + \zeta, y)$, donde ζ es un vector de ruido Gaussiano generado aleatoriamente creando cada una de sus coordenadas a partir de una distribución Gaussiana con esperanza 0 y varianza igual a 0,16. Nos referimos a éste conjunto de datos como sintético en tablas de resultados experimentales que aparecen más adelante en esta sección. No usamos ningún núcleo en éste conjunto de datos para ninguno de los experimentos.

El segundo conjunto de datos utilizado en nuestros experimentos es un corpus del habla natural continua para la tarea de clasificación de fonemas. Los datos que hemos utilizado son un subconjunto del conjunto de datos fonético-acústicos TIMIT, que como ya se mencionó es un corpus fonéticamente transcrito, del habla continua y de alta calidad y que se describe con más detalle en el *Apéndice A*, [Lamel et al., 1986]. Se extrajeron los coeficientes cepstrales en las frecuencias Mel[3]), de la expresión hablada junto con su primera y segunda derivada en forma estándar[4], y cada vector de características se generó a partir de los 5 vectores MFCC adyacentes (con superposición). El corpus TIMIT se divide en un conjunto de entrenamiento y uno de prueba de tal manera que no hay hablantes del conjunto de entrenamiento que aparezcan también en el conjunto de pruebas (independiente del hablante). Se seleccionaron aleatoriamente 2000 vectores de características para entrenamiento y 500 vectores de características de prueba por cada uno de los 40 fonemas. Normalizamos los datos para tener media cero y varianza unitaria. En todos los experimentos con este conjunto de datos utilizamos un núcleo *función de base radial* (RBF[5]) con $\sigma = 0{,}5$.

Entrenamos y probamos las versiones *en línea* y *de procesamiento por lotes* del algoritmo en los dos conjuntos de datos. Para demostrar los beneficios del uso y aprovechamiento de la estructura jerárquica sugerida, también entrenamos y evaluamos predictores multiclase estándar que ignoran por completo la estructura jerárquica de fonemas. Estos clasificadores fueron

[3]Mejor conocidos como MFCC del inglés Mel Frequency Cepstral Coefficients.
[4]Basándonos en el estándar ETSI para reconocimiento de voz distribuido [ES-202-211].
[5]Del inglés Radial Basis Function.

entrenados mediante el mismo algoritmo, pero con una versión "plana" de la estructura jerárquica de fonemas. El error inducido por el árbol acumulado y el porcentaje de errores multiclase para cada experimento se resumen en las Tablas 3.2 y 3.3 para los experimentos en línea y en las Tablas 3.4, 3.5 y 3.6 para los experimentos en la configuración de procesamiento por lotes por lotes. Las tablas encabezadas con *"usando la estructura jerárquica"* se refieren al desempeño del algoritmo entrenado con conocimiento de dicha estructura, mientras que las tablas tituladas como *"sin estructura jerárquica"* se refieren a los resultados del clasificador entrenado con los fonemas en su versión "plana" sin conocimiento de la jerarquía.

Los resultados indican claramente que el uso y aprovechamiento de la estructura jerárquica es ventajosa para alcanzar valores pequeños de error inducido por el árbol. En todos los experimentos, tanto *en línea* como *por lotes* (batch), el clasificador jerárquico de fonemas logró una disminución en los valores de *error inducido por el árbol* con respecto a su contraparte "plana". Por otra parte, en la mayoría de los experimentos, el *error multiclase* del algoritmo también es menor que el *error del predictor multiclase* correspondiente, aunque este último se entrenó expresamente para minimizar el error. Consideramos que este comportamiento es un ejemplo de que el empleo de una estructura jerárquica de fonemas puede ser útil incluso cuando el objetivo no es necesariamente la minimización de un error basado en árbol.

Un examen más detallado de los resultados demuestra que el clasificador jerárquico de fonemas tiende a tolerar pequeños errores inducidos por el árbol, y al mismo tiempo evita los más grandes. El algoritmo por lotes tiende a cometer errores "pequeños" en la predicción de fonemas padre o hermano del fonema correcto. Por otro lado el algoritmo rara vez elige un fonema o un grupo de fonemas que se encuentra en una parte totalmente diferente del árbol, lo que evita grandes errores inducidos por el árbol. En la tarea de la clasificación de fonemas, el algoritmo muy rara vez ofrece una predicción $\hat{y}$ tal que $\gamma(y, \hat{y}) = 9$, en tanto que los errores del predictor multiclase están distribuidos uniformemente.

Concluimos los experimentos con una comparación del algoritmo jerárquico con una construcción común de los clasificadores jerárquicos (véase [Koller y Sahami, 1997]), donde los clasificadores independientes se aprenden y son aplicados en cada vértice interno de la jerarquía de manera inde-

Sin la estructura jerárquica		
Conjunto de datos	**Error inducido por el árbol**	**Error multiclase**
Fonemas	1.67	**38.6**
Sintéticos	1.43	52.3

Tabla 3.2: Resultados experimentales del algoritmo *en línea* sin usar la estructura jerárquica de fonemas.

Usando la estructura jerárquica		
Conjunto de datos	**Error inducido por el árbol**	**Error multiclase**
Fonemas	**1.56**	39.2
Sintéticos	**0.76**	**45.3**

Tabla 3.3: Resultados experimentales del algoritmo *en línea* usando la estructura jerárquica de fonemas.

pendiente. Para comparar los dos enfoques, aprendemos un predictor multiclase en cada vértice interno de la jerarquía de árbol. Cada uno de tales clasificadores envía o direcciona un vector de características de entrada a uno de sus hijos. Formalmente decimos que, para cada vértice interno v de $\mathcal{T}$ entrenamos un clasificador f_v usando el conjunto de entrenamiento $S_v = \{(\mathbf{x}_i, u_i) \,|u_i \in \mathcal{P}(y_i), v = \mathcal{A}(u_i), (\mathbf{x}_i, y_i) \in S\}$. Dado un vector de características de prueba $\mathbf{x}$, su fonema predicho es la hoja $\hat{y}$ del árbol, de tal manera que para cada $u \in \mathcal{P}(\hat{y})$ y su padre v tenemos que $f_v(\mathbf{x}) = u$. En otras palabras, para emitir una predicción comenzamos en el vértice raíz y avanzamos hacia una de las hojas avanzando a partir de un vértice v hasta $f_v(\mathbf{x})$. Hacemos referencia a este modelo de clasificación jerárquica en la Tabla 3.4 marcándolo como *modelo ambicioso*. En todos los experimentos, el algoritmo por lotes supera claramente al modelo ambicioso. Este experimento subraya la utilidad del enfoque presentado que toma decisiones globales, en contraste con las decisiones locales de la construcción del algoritmo ambicioso. De hecho, cualquier error individual de predicción en cualquiera de los vértices en la ruta hacia el fonema correcto impone un error de predicción global.

Usando el modelo ambicioso				
	Error inducido por el árbol		Error multiclase	
Conjunto de datos	**Batch simple**	**Batch**	**Batch simple**	**Batch**
Fonemas	3.46	2.72	75.0	57.1
Sintético	0.63	0.43	34.3	33.2

TABLA 3.4: Resultados experimentales del algoritmo *de procesamiento por lotes* usando el modelo ambicioso.

Sin la estructura jerárquica				
	Error inducido por el árbol		Error multiclase	
Conjunto de datos	**Batch simple**	**Batch**	**Batch simple**	**Batch**
Fonemas	2.22	1.34	47.52	43.0
Sintético	0.16	0.14	11.2	8.3

TABLA 3.5: Resultados experimentales del algoritmo *de procesamiento por lotes* sin usar la estructura jerárquica de fonemas.

Usando la estructura jerárquica				
	Error inducido por el árbol		Error multiclase	
Conjunto de datos	**Batch simple**	**Batch**	**Batch simple**	**Batch**
Fonemas	**1.96**	**1.42**	**47.64**	**39.81**
Sintético	**0.06**	**0.08**	**4.42**	**5.31**

TABLA 3.6: Resultados experimentales del algoritmo *de procesamiento por lotes* usando la estructura jerárquica de fonemas.

Capítulo 4
Alineamiento de fonemas

Resumen:

Proponemos el uso de una nueva técnica para alinear una secuencia de fonemas proveniente de una expresión hablada, con su respectiva señal acústica. A diferencia de los enfoques basados en MOMs, el método emplea un procedimiento de aprendizaje discriminatorio en el cual la fase de aprendizaje está estrechamente ligada a la tarea de alineamiento que se realiza manualmente.

La función de alineamiento que presentamos se basa en el mapeo de las representaciones simbólico-acústicas de la expresión hablada en la entrada junto con la alineación deseada, a un espacio vectorial abstracto. Se propone realizar un mapeo específico a un espacio vectorial que utilize características estándar del habla (por ejemplo las distancias espectrales), así como los resultados a la salida de un clasificador de fonemas que se describe mas adelante en este trabajo.

Dicha función de alineamiento está basada en las técnicas utilizadas por los métodos de margen amplio para la predicción de secuencias completas. Se describe un algoritmo iterativo simple para el entrenamiento de la función de alineamiento y se discuten sus propiedades. Los experimentos realizados con el corpus TIMIT muestran que el método presentado compite con otras aproximaciones consideradas actualmente parte del estado del arte.

4.1. Introducción

Hoy en día el desarrollo de tecnologías del habla depende fuertemente de metodologías basadas en el uso corpus, y por lo tanto de la disponibilidad de buenos corpus de habla. Para que un corpus sea realmente útil para el desarrollo de sistemas de de reconocimiento automático del habla, además de la grabación de la expresiones del habla o frases, debe contener información sobre su contenido (etiquetado) y sobre el alineamiento en el tiempo entre el etiquetado y la expresión hablada (segmentación). Los fonos suelen considerarse las unidades mínimas del habla, por cuya concatenación cualquier otra unidad del habla (sílaba, palabra, frase, etc.) puede ser construida. No es de extrañar, que la segmentación fonética y el etiquetado sean muy deseables y útiles en un corpus de habla. Hasta ahora la manera más precisa de etiquetar un corpus es de forma manual. Sin embargo, el etiquetado fonético manual, y muy especialmente la segmentación, son muy costosos y requieren mucho tiempo y esfuerzo [Cosi et al., 1991; Kacur et al., 2008].

En el área del reconocimiento automático del habla, el uso de los MOMs ha evitado la necesidad de la segmentación fonética manual. Los MOMs producen una segmentación, que aunque menos precisa que la segmentación manual, parece ser lo suficientemente precisa como para entrenar dichos modelos, ya que el entrenamiento para los MOMs es un proceso de obtención de promedios que tiende a suavizar los errores en la segmentación. Algunas investigaciones aseguran que el RAH debe beneficiarse del mejoramiento en la precisión de la segmentación de fonemas [Baghai-Ravary, 2010], principalmente para realizar las tareas de entrenamiento [Malfrère et al., 1998] y decodificación [Mitchell et al., 1995].

El método más frecuente para la segmentación automática de fonemas es modificar al reconocedor para adaptarlo a la tarea de la segmentación fonética [Gubrynowicz y Wrzoskowicz, 1993; Ljolje y Riley, 1993; Cox et al., 1998]. La principal modificación necesaria consiste en darle al reconocedor la transcripción fonética de la frase a segmentar (que se considera conocida) mediante la construcción de una gramática para esa transcripción y la realización de una alineación forzada. En otras investigaciones se ha intentado el uso de diferentes características y técnicas, entre ellas se pueden mencionar a la amplitud [Farhat et al., 1993; Kabré et al., 1991] el contorno de energía de tiempo corto [Gholampour y Nayebi, 1998], la energía en diferentes

bandas de frecuencia [Farhat et al., 1993; Kabré et al., 1991] las funciones de variación espectral [Mitchell et al., 1995], el contorno *f0* [Saito, 1998], y características sobre la base de modelos auditivos [Yarrington et al., 1995].

Otras técnicas que han sido utilizadas incluyen la pre-segmentación en segmentos de espectro estable [Farhat et al., 1993; Kabré et al., 1991] o la pre-segmentación de varios niveles representados como dendogramas [Gholampour y Nayebi, 1998], posiblemente seguido de una alineación de segmentos y fonemas [Farhat et al., 1993], la detección basada en reglas de límites fonéticos [Houben, 1989; Hatazaki et al., 1989], las redes neuronales como detectores de bordes fonéticos [Vorstermanst et al., 1995], el alineamiento de la expresión hablada con la realización de la misma expresión producida por un sintetizador de voz [Malfrère et al., 1998; Yarrington et al., 1995] y el alineamiento dinámico del tiempo[1] [Saito, 1998].

En algunas investigaciones se han tratado de combinar modelos como los MOMs, con otras características y técnicas, pero la mayor parte del trabajo previo respecto a la segmentación de fonemas se ha centrado en los modelos generativos de la señal de voz utilizando solamente MOMs. Véanse por ejemplo Huggins-Daines y Rudnicky [2006]; Hosom [2002]; Brugnara et al. [1993]; Toledano et al. [2003]. Como se mencionó anteriormente, a pesar de su popularidad, los enfoques basados en tales modelos tienen varios inconvenientes, tales como la convergencia del procedimiento de maximización de la esperanza a máximos locales y los efectos del sobreajuste debido a la gran cantidad de parámetros. En este trabajo se propone un enfoque alternativo para la alineación de fonemas que se basa en investigaciones recientes sobre aprendizaje supervisado discriminatorio.

La ventaja de los algoritmos de aprendizaje discriminatorios derivan del hecho de que la función utilizada durante la fase de aprendizaje está estrechamente ligada con la tarea de decisión que se necesita llevar a cabo. Además, existe evidencia tanto teórica como empírica de que los algoritmos discriminatorios de aprendizaje probablemente superan a los modelos generativos en la misma tarea (cf. [Cristianini y Shawe-Taylor, 2000] y [Vapnik, 1998]). Uno de los algoritmos de aprendizaje discriminatorio más conocidos son las máquinas de vectores de soporte(MVS), que han sido utilizadas con éxito en algunas aplicaciones de procesamiento del habla [Salomon et al., 2002; Keshet

[1]Mejor conocido como DTW por sus siglas en Inglés Dynamic Time Warping.

et al., 2001]. El algoritmo de MVS clásico está diseñado para tareas simples de decisión, tales como la clasificación binaria y la regresión. Por lo tanto, su aprovechamiento en sistemas de procesamiento de voz hasta el momento también se ha restringido a tareas de decisión simple como la clasificación de fonemas. El problema de alineamiento de fonemas es más complicado, dado que es necesario predecir una secuencia *de tiempos de inicio* de fonemas en lugar de un solo valor numérico.

El reto principal en esta parte del trabajo es extender la noción de aprendizaje discriminatorio para la compleja tarea de alineación de fonemas. El método propuesto se basa en los recientes avances en funciones núcleo y clasificadores de margen amplio para secuencias (véase [Shalev-Shwartz et al., 2004]) cuyos fundamentos se establecen principalmente en el trabajo pionero de Vapnik [Vapnik, 1998]. La función de alineamiento sugerida se basa en el mapeo de la señal de voz y su representación en fonemas junto con el alineamiento deseado, en un espacio vectorial abstracto. Basándose en las técnicas utilizadas para aprendizaje con MVS, la función de alineamiento entrega resultados a un clasificador en este espacio vectorial, que tiene por objeto separar los alineamientos correctos de los incorrectos. Se describe el algoritmo iterativo para el entrenamiento de la función de alineación y se discuten sus propiedades. Los experimentos realizados con el corpus TIMIT muestran que el método que se presenta mejora el desempeño en comparación con aquellos basados en el enfoque MOM. En particular usamos los trabajos de Brugnara et al. [1993] y Hosom [2002] como línea base y punto de comparación. Esta parte del trabajo está organizada de la siguiente manera. En la *Sección* 2 se introduce formalmente el problema de alineamiento de fonemas. El método de entrenamiento se describe en la *Sección* 3. Por último en la *Sección* 4 se presentan los resultados experimentales.

4.2. El problema del alineamiento de fonemas

En esta sección se describe formalmente el problema del alineamiento, para su solución contamos con una expresión hablada junto con una representación fonética de la expresión. Nuestro objetivo es alinear la señal de voz con su representación fonética. Los coeficientes cepstrales en las frecuencias

Mel[2], junto con sus primeras y segundas derivadas se extraen de la expresión hablada[3]. Denotamos al dominio de los vectores de características acústicas como $\mathcal{X} \subset \mathbb{R}^d$. Las características acústicas de la señal de voz se representan mediante una secuencia de vectores $\overline{\mathbf{x}} = (\mathbf{x}_1, ..., \mathbf{x}_T)$, en donde $\mathbf{x}_t \in \mathcal{X}$ para todo $1 \leq t \leq T$. La representación fonética de una expresión se define como una cadena de símbolos de fonemas y formalmente, representamos a cada fonema como $p \in \mathcal{P}$, donde $\mathcal{P}$ es el conjunto de símbolos de fonemas.

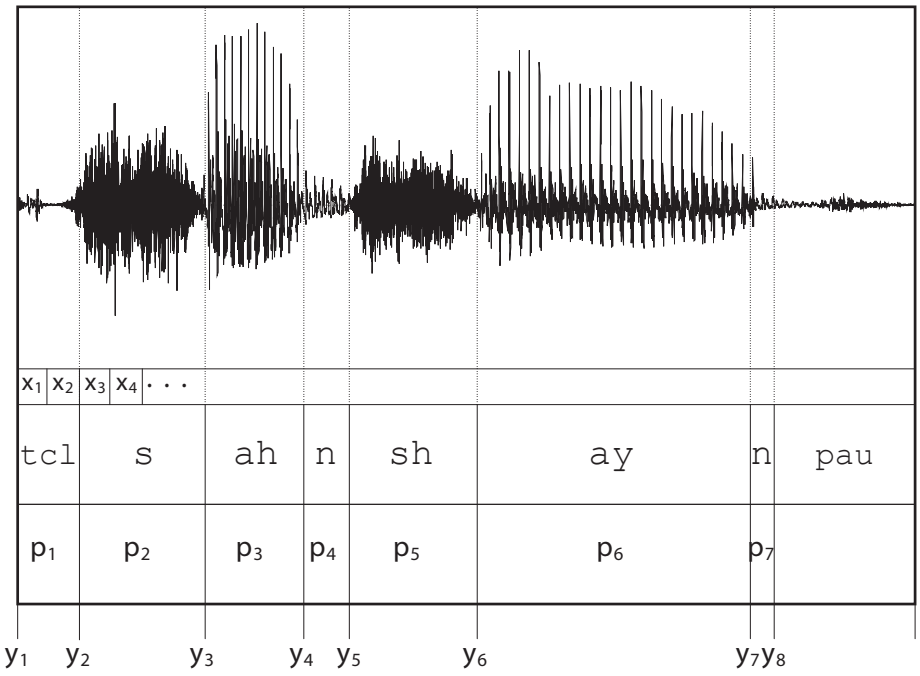

Figura 4.1: Señal de voz para la palabra "*sunshine*" tomada del corpus TIMIT, etiquetada con su respectiva secuencia de fonemas $(p_1, p_2, ..., p_7)$ y su correspondiente secuencia de tiempos de inicio $(y_1, y_2, ..., y_8)$.

Por lo tanto, la representación fonética de una expresión hablada consiste en una secuencia de valores de fonema $\overline{p} = (p_1, ..., p_k)$. Hay que tener cuenta que el número de fonemas varía de una expresión a otra y por lo tanto k no

[2]Mejor conocidos como MFCC del inglés Mel-frequency cepstrum coefficients.
[3]Basándonos en el estándar ETSI para reconocimiento de voz distribuido [ES-202-211].

es un valor fijo. En resumen, una entrada para la tarea de alineamiento es un par $(\overline{\mathbf{x}}, \overline{p})$, donde $\overline{\mathbf{x}}$ es una representación acústica de la señal de voz y $\overline{p}$ es una representación fonética de la misma señal. Un alineamiento entre las representaciones acústicas y fonéticas de una expresión hablada es una secuencia de los tiempos de inicio $\overline{y} = (y_1, ..., y_k)$ en donde $y_i \in \mathbb{N}$ es el tiempo de inicio del fonema i (medido como un número de trama) en la señal acústica. Por lo tanto, cada fonema i se inicia en la trama y_i y termina en la trama $y_{i+1} - 1$. Un ejemplo de la notación descrita se ilustra en la Fig. 4.1.

Está claro que hay diferentes maneras de pronunciar una misma expresión, diferentes hablantes tienen distintos acentos y tienden a hablar a un ritmo diferente. Nuestro objetivo es entrenar una función de alineamiento que sea capaz de predecir los tiempos de inicio reales de los fonemas a partir de la señal de voz y su representación fonética. En este punto se discuten los enfoques generativos comúnmente usados para el alineamiento de fonemas. En el paradigma generativo, se asume que la señal de voz $\overline{\mathbf{x}}$ se genera a partir de la secuencia de fonemas $\overline{p}$ y de la secuencia de sus tiempos de inicio $\overline{y}$, con base en una función de probabilidad $P(\overline{\mathbf{x}} \mid \overline{p}, \overline{y})$. La predicción del tipo Máximo a Posteriori (MAP) es por lo tanto,

$$\overline{y}' = \underset{\overline{y}}{\operatorname{argmax}}\ P(\overline{y} \mid \overline{\mathbf{x}}, \overline{p}) = \underset{\overline{y}}{\operatorname{argmax}}\ P(\overline{y} \mid \overline{p})\ P(\overline{\mathbf{x}} \mid \overline{p}, \overline{y}),$$

donde la última igualdad se deduce de la regla de Bayes. Dicho de otra manera, la predicción de $\overline{y}'$ se basa en dos funciones de probabilidad, una probabilidad a priori $P(\overline{y} \mid \overline{p})$ y una probabilidad posterior $P(\overline{\mathbf{x}} \mid \overline{p}, \overline{y})$. Para facilitar el cálculo eficiente de $\overline{y}'$, los modelos generativos asumen que las funciones de probabilidad pueden todavía descomponerse en funciones básicas de probabilidad. Por ejemplo, en el marco de los MOMs comúnmente se asume que,

$$P(\overline{\mathbf{x}} \mid \overline{p}, \overline{y}) = \prod_i \prod_t P(\mathbf{x}_t \mid p_i),$$

y además,

$$P(\overline{y} \mid \overline{p},) = \prod_i P(\ell_i \mid \ell_{i-1}, p_i, p_{i-1}),$$

donde $\ell_i = y_{i+1} - y_i$ es la longitud del $i-$ésimo fonema de acuerdo con $\overline{y}$. Todos estos supuestos aun cuando ayudan a simplificar el problema, también conducen a un modelo que es inadecuado para el propósito de generar expresiones del habla naturales. Sin embargo, la probabilidad de la secuencia de tiempos de inicio de los fonemas, dada la señal de voz y las secuencias de fonemas, se utiliza como una valoración de la calidad de la secuencia de alineamiento.

La fase de aprendizaje en los MOMs, tiene como propósito determinar las funciones de probabilidad básicas a partir de un conjunto de ejemplos de entrenamiento. El objetivo del entrenamiento es encontrar las funciones $P(\mathbf{x}_t \mid p_i)$ y $P(\ell_i \mid \ell_{i-1}, p_i, p_{i-1})$ que maximizan la verosimilitud del conjunto de entrenamiento. Teniendo en cuenta estas funciones, la predicción $\overline{y}'$ se calcula en la fase de inferencia, que se puede realizar de manera eficiente mediante el uso de programación dinámica. En esta parte del trabajo describimos y analizamos un enfoque alternativo en el que la etapa de aprendizaje está estrechamente relacionada con la tarea de decisión que el algoritmo debe realizar. En lugar de trabajar con las funciones de probabilidad suponemos la existencia de un conjunto predefinido de funciones base de alineamiento, $\{\phi_j\}_{j=1}^{n}$. Cada función base toma la forma $\phi_j : \mathcal{X}^* \times (\mathcal{P}\times\mathbb{N})^* \longrightarrow \mathbb{R}$. Por lo tanto, la entrada de cada una de estas funciones base es una representación fonético-acústica $(\overline{\mathbf{x}}, \overline{p})$, junto con una alineación tentativa $\overline{y}$. La función base devuelve un valor escalar, que representa intuitivamente el valor de confianza en el alineamiento $\overline{y}$ sugerido. Si denotamos por $\phi(\overline{\mathbf{x}}, \overline{p}, \overline{y})$ al vector en $\mathbb{R}^n$ cuyo j-ésimo elemento es $\phi_j(\overline{\mathbf{x}}, \overline{p}, \overline{y})$, entonces las funciones de alineamento que usaremos tienen la forma

$$f(\overline{\mathbf{x}}, \overline{p}) = \underset{\overline{y}}{\operatorname{argmax}}\, \mathbf{w} \cdot \phi(\overline{\mathbf{x}}, \overline{p}, \overline{y}), \tag{4.1}$$

en donde $\mathbf{w} \in \mathbb{R}^n$ es un vector de pesos de importancia que debe ser entrenado. Es decir, f devuelve una *secuencia tentativa de alineamiento* mediante la maximización de la suma ponderada de las puntuaciones devueltas por cada función base ϕ_j. Hay que tomar en cuenta que el número de posibles secuencias de alineamiento es exponencialmente grande. Sin embargo, como en el caso generativo, si las funciones base ϕ_j pueden descomponerse, entonces la versión optimizada de la Ec. 4.1 puede calcularse de manera eficiente usando un procedimiento de programación dinámica.

Como se mencionó, nos interesa entrenar a la función f a partir de un cierto número de ejemplos. Cada ejemplo se compone de la representación acústica y fonética de una expresión $(\overline{\mathbf{x}}, p)$, junto con el alineamiento correcto entre ellos $\overline{y}$. Sea $\overline{y}' = f(\overline{\mathbf{x}}, p)$ el alineamiento sugerido por f. Denotamos como $\gamma(\overline{y}, \overline{y}')$ al costo de predecir la alineación $\overline{y}'$ en donde la verdadera alineación es $\overline{y}$. Formalmente $\gamma : (\mathbb{N} \times \mathbb{N})^* \longrightarrow \mathbb{R}$ es una función que recibe dos alineaciones y devuelve un escalar que representa el costo de la predicción de $\overline{y}'$ cuando la verdadera alineación es $\overline{y}$. Asumiendo que $\gamma(\overline{y}, \overline{y}') \geq 0$ y que $\gamma(\overline{y}, \overline{y}) = 0$. Un ejemplo de la función de costo es,

$$\gamma(\overline{y}, \overline{y}') = \frac{1}{|\overline{y}|} \sum_{i=1}^{|\overline{y}|} |y_i - y'_i| \,. \tag{4.2}$$

La Ec. 4.2 o *ecuación de costo*, representa el promedio de las diferencias absolutas entre la alineación que se predice y la alineación correcta. El objetivo del proceso de aprendizaje consiste en encontrar una función f de alineación que logre un bajo costo para los ejemplos "ocultos". Por ello, en la siguiente sección se muestra cómo utilizar el conjunto de entrenamiento con el fin de encontrar una función de alineación f que con alta probabilidad, logre un pequeño costo dado el conjunto de entrenamiento, así como con los ejemplos ocultos.

4.3. La etapa de entrenamiento

En esta sección presentamos los detalles de la aproximación discriminatoria propuesta. Recordemos que su construcción se basa en un conjunto de funciones base de alineamiento $\{\phi_j\}_{j=1}^{n}$ que mapean la representación fonético-acústica de una expresión del habla, así como un alineamiento sugerido a un espacio vectorial abstracto. Comenzamos esta sección introduciendo un conjunto específico de funciones base que consideramos muy adecuado para el problema de alineación de fonemas, a continuación, se describe un sencillo procedimiento iterativo para encontrar el vector de pesos $\mathbf{w}$. El papel de $\mathbf{w}$ es calificar a los alineamientos posibles para una expresión de entrada de tal manera que la alineación correcta sea aquella que alcance la calificación más alta o superior.

4.3.1. Funciones base de alineamiento

Utilizamos siete diferentes funciones base ($n = 7$). Estas funciones base se utilizan para la definición de nuestra función de alineamiento tal como en la Ec. 4.1. Para facilitar una evaluación eficaz de $f(\overline{\mathbf{x}}, \overline{p})$ se deben cumplir las restricciones estructurales de las funciones base. A continuación, se describen estas funciones base, prestando especial atención a sus propiedades de descomposición, que más tarde nos permitirán evaluar de forma eficiente $f(\overline{\mathbf{x}}, \overline{p})$, utilizando un procedimiento de programación dinámica. Vale la pena mencionar que el mismo tipo de suposiciones estructurales se asumen en los enfoques basados en MOMs.

Las primeras cuatro funciones base tienen como objetivo capturar las transiciones entre fonemas. Estas funciones se basan en la distancia entre tramas de la señal acústica en los dos lados del límite del fonema, según lo sugerido por el alineamiento $\overline{y}$. La medida de distancia que empleamos, y que denotamos como d, es la distancia Euclideana entre vectores de características. Nuestra suposición es que si dos tramas $\mathbf{x}_t$ y $\mathbf{x}_{t'}$, se derivan del mismo fonema, entonces la distancia $d(\mathbf{x}_t, \mathbf{x}_{t'})$ debe ser menor que en el caso en que las dos tramas se derivan de diferentes fonemas. Definimos formalmente a las primeras 4 funciones base como:

$$\phi_s(\overline{\mathbf{x}}, \overline{p}, \overline{y}) = \sum_{i=1}^{|\overline{y}|} d(\mathbf{x}_{y_i - s}, \mathbf{x}_{y_i + s}), \qquad \text{para } s \in \{1, 2, 3, 4\}. \tag{4.3}$$

Si $\overline{y}$ es el alineamiento correcto entonces las distancias entre tramas a través de los puntos de cambio de fonema probablemente serán grandes. Por el contrario, en un alineamiento incorrecto es probable que al comparar las tramas del mismo fonema, a menudo obtengamos distancias pequeñas.

La quinta función base que utilizamos se basa en el clasificador de fonemas tipo *framewise* que se describe en Dekel et al. [2004]. Formalmente, para cada fonema $p \in \mathcal{P}$ y trama $\mathbf{x} \in \mathcal{X}$, existe un valor de confianza de que el fonema p es pronunciado en la trama $\mathbf{x}$ y que se denota como $g_p(\mathbf{x})$. La función base resultante mide el valor de confianza acumulado de la señal de voz completa,

dada la secuencia de fonemas, y sus tiempos de inicio, como sigue

$$\phi_5(\overline{\mathbf{x}}, \overline{p}, \overline{y}) = \sum_{i=1}^{|\overline{p}|} \sum_{t=y_i}^{y_{i+1}-1} g_{p_i}(\mathbf{x}_t). \tag{4.4}$$

La sexta función base asigna puntajes a las alineaciones basándose en la duración del fonema. A diferencia de las anteriores funciones base, esta función es ajena a la señal de voz en sí. Simplemente examina la longitud de cada fonema, como se sugiere en $\overline{y}$, en comparación con la típica duración requerida para pronunciar este fonema. Expresada formalmente, decimos que

$$\phi_6(\overline{\mathbf{x}}, \overline{p}, \overline{y}) = \sum_{i=1}^{|\overline{y}|-1} \mathcal{N}(y_{i+1} - y_i; \hat{\mu}_{p_i}, \hat{\sigma}_{p_i}), \tag{4.5}$$

donde $\mathcal{N}$ es una función de densidad de probabilidad Normal con media $\hat{\mu}_p$ y desviación estándar $\hat{\sigma}_p$. En los experimentos llevados a cabo, hemos estimado $\hat{\mu}_p$ y $\hat{\sigma}_p$ a partir del conjunto completo de entrenamiento del corpus TIMIT.

La última función base aprovecha los supuestos acerca de la velocidad del habla de un "hablante común". Intuitivamente la gente habla a una tasa prácticamente constante y por lo tanto, una secuencia de alineamiento en la que se cambie bruscamente la velocidad del habla es muy probablemente incorrecta. Formalmente decimos que siendo $\hat{\mu}_p$ la duración promedio requerida para pronunciar el fonema $p-$ésimo y denotando como r_i a la velocidad relativa del habla, entonces $r_i = \dfrac{(y_{i+1} - y_i)}{\hat{\mu}_p}$. Es decir r_i es la proporción entre la longitud real del fonema p_i tal como lo sugiere $\overline{y}$, y su duración promedio. Lo cual supone que la velocidad relativa del habla cambia con lentitud en el tiempo. En la práctica, la proporcion en la velocidad del habla a menudo difiere de un hablante a otro y para una expresión dada. Medimos el cambio local en la velocidad de habla como $(r_i - r_{i-1})^2$ y por ello definimos la función base ϕ_7 como la suma acumulada de los cambios en la velocidad de habla,

$$\phi_7(\overline{\mathbf{x}}, \overline{p}, \overline{y}) = \sum_{i=2}^{|\overline{y}|-1} (r_i - r_{i-1})^2. \tag{4.6}$$

Concluimos la descripción de las funciones base para alineación con un análisis de la evaluación práctica de la función de alineamiento f. Hay que recordar que el cálculo de f requiere resolver el problema de optimización, $f(\overline{\mathbf{x}}, \overline{p}) = \text{argmax}_{\overline{y}}\, \mathbf{w} \cdot \phi(\overline{\mathbf{x}}, \overline{p}, \overline{y})$. Una búsqueda directa de su maximizador no es factible ya que el número secuencias de alineamiento $\overline{y}$ posibles es exponencial en la longitud de la secuencia. Afortunadamente, las funciones base que hemos presentado se pueden descomponer y por lo tanto la mejor secuencia de alineamiento se puede encontrar en tiempo polinomial usando programación dinámica (en forma similar al algoritmo de Viterbi que a menudo se implementa en la aproximación basada en MOMs [Rabiner y Juang, 1993]).

4.3.2. El algoritmo de entrenamiento

Pasando a la descripción del algoritmo iterativo de entrenamiento, para realizar la tarea de alineamiento de fonemas, recordemos que un algoritmo de aprendizaje supervisado recibe como entrada un conjunto de entrenamiento $S = \{(\overline{\mathbf{x}}_1, \overline{p}_1, \overline{y}_1), ..., (\overline{\mathbf{x}}_m, \overline{p}_m, \overline{y}_m)\}$ y devuelve un vector de pesos $\mathbf{w}$ que define la función de alineación f dada por la Ec. 4.1. Al igual que en el algoritmo con MVS para la clasificación binaria (que revisamos como parte de los antecedentes), nuestro enfoque para la elección del vector de pesos $\mathbf{w}$ se basa en la idea de la separación con margen amplio. Sin embargo, en este caso, los alineamientos solo pueden ser correctos o incorrectos. En cambio, la función de costo $\gamma(\overline{y}, \overline{y}')$ se utiliza para evaluar la calidad de los alineamientos. Es decir, no tiene por objeto separar los alineamientos correctos de los incorrectos, sino más bien tratar de calificarlos de acuerdo a su calidad. En teoría, el enfoque que presentamos puede ser descrito como un procedimiento de dos pasos.

En primer lugar, construimos un vector $\phi(\overline{\mathbf{x}}_i, \overline{p}_i, \overline{y}')$ en el espacio vectorial $\mathbb{R}^n$ basados en cada ejemplo $(\overline{\mathbf{x}}_i, \overline{p}_i)$ en el conjunto de entrenamiento S y cada posible alineamiento $\overline{y}'$. En segundo lugar, encontramos un vector $\mathbf{w} \in \mathbb{R}^n$, tal que la proyección de los vectores en $\mathbf{w}$ califica a los vectores (les otorga una calificación) construidos en la primera etapa de acuerdo con su calidad. Formalmente, para cada instancia $(\overline{\mathbf{x}}_i, \overline{p}_i)$ y para cada posible alineamiento sugerido $\overline{y}'$, la siguiente restricción debe mantenerse,

$$\mathbf{w} \cdot \phi(\overline{\mathbf{x}}_i, \overline{p}_i, \overline{y}_i) - \mathbf{w} \cdot \phi(\overline{\mathbf{x}}_i, \overline{p}_i, \overline{y}') \geq \gamma(\overline{y}_i, \overline{y}') - \xi_i, \tag{4.7}$$

donde ξ_i es una variable de holgura no negativa e indica la pérdida ocurrida hasta el $i-$ésimo ejemplo. La solución para el problema usando MSV es el vector de pesos $\mathbf{w}$ que minimiza la función objetivo $\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i$ y al mismo tiempo satisface todas las restricciones en la Ec. 4.7. El parámetro C sirve para el ajuste del balance entre complejidad y precisión (véase Cristianini y Shawe-Taylor [2000]).

En la práctica, el procedimiento anterior de dos pasos no se puede aplicar directamente dado el número exponencialmente grande de restricciones. Para superar este obstáculo, se describe un sencillo procedimiento iterativo para encontrar $\mathbf{w}$. El algoritmo iterativo primero construye una secuencia de vectores de pesos $\mathbf{w}_0, \mathbf{w}_1, ..., \mathbf{w}_m$. El primer vector de pesos inicia como el vector cero, $\mathbf{w}_0 = 0$. En la iteración i del algoritmo, utilizamos el $i-$ésimo ejemplo del conjunto de entrenamiento junto con el vector de pesos anterior $\mathbf{w}_i$, para definir el siguiente vector de pesos $\mathbf{w}_{i+1}$. Sea $\overline{y}'$ la secuencia de alineación predicha para el $i-$ésimo ejemplo de acuerdo con $\mathbf{w}_i$. Establecemos el siguiente vector de pesos $\mathbf{w}_{i+1}$ para que sea el minimizador del siguiente problema de optimización,

$$\min_{\mathbf{w}\in\mathbb{R}^n,\xi\geq 0} \quad \frac{1}{2}\|\mathbf{w}-\mathbf{w}_i\|^2 + C\xi, \tag{4.8}$$

tal que,

$$\mathbf{w}\cdot\phi(\overline{\mathbf{x}},\overline{p},\overline{y}) - \mathbf{w}\cdot\phi(\overline{\mathbf{x}},\overline{p},\overline{y}') \geq \sqrt{\gamma(\overline{y},\overline{y}')} - \xi.$$

Este problema de optimización puede considerarse como una variación del problema de optimización con MVS, con tres importantes diferencias. En primer lugar, sustituimos el número de restricciones de la Ec. 4.7 con una única restricción. Esta restricción se basa en la alineación predicha $\overline{y}'$ de acuerdo con el vector de pesos $\mathbf{w}_i$ anterior. En segundo lugar, reemplazamos el término $\|\mathbf{w}\|^2$ en la función objetivo de la MVS con el término $\|\mathbf{w}-\mathbf{w}_i\|^2$. Intuitivamente, sabemos que lo deseable es minimizar la pérdida de $\mathbf{w}$ en el ejemplo actual, como por ejemplo la variable ξ de holgura, y al mismo tiempo queremos permanecer lo más cerca posible a nuestro vector de pesos $\mathbf{w}_i$ anterior. Para ello, por último, reemplazamos $\gamma(\overline{y},\overline{y}')$ con $\sqrt{\gamma(\overline{y},\overline{y}')}$. Se puede demostrar (véase Crammer et al. [2006]) que la solución al problema de optimización anterior es,

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \text{mín}\left\{\frac{\ell}{\|a\|^2}, C\right\} a$$

en donde

$$a = \phi(\overline{\mathbf{x}}_i, \overline{p}_i, \overline{y}_i) - \phi(\overline{\mathbf{x}}_i, \overline{p}_i, \overline{y}') \text{ y } \ell = \text{máx}\{(\gamma(\overline{y}_i, \overline{y}'))^{\frac{1}{2}} - \mathbf{w}_i \cdot a, 0\}.$$

El procedimiento iterativo anterior nos entrega una secuencia de vectores de pesos. Se puede mostrar que al menos una de las funciones de alineación resultante tiene buenas propiedades de generalización (véase Cesa-Bianchi et al. [2004]). Para encontrar una función de alineamiento que se pueda generalizar, se calcula el costo promedio de cada función de alineamiento en un conjunto de validación y elegimos aquella que logra los mejores resultados.

4.4. Resultados experimentales

En esta sección se presentan los resultados experimentales que muestran la precisión del algoritmo presentado. Para validar la eficacia del método propuesto se realizaron experimentos con el corpus TIMIT que fue construido para entrenar y evaluar reconocedores de fonemas independientes del hablante. En todos los experimentos, el sistema discriminatorio de referencia y el sistema basado en MOMs fueron entrenados con las expresiones habladas en el corpus TIMIT y con un conjunto de fonemas como se define en Lee y Hon [1989].

En primer lugar dividimos la parte del corpus TIMIT elaborado para fines de entrenamiento en tres partes disjuntas conteniendo 500, 100 y 3090 frases, excluyendo las expresiones habladas en los subconjuntos $SA1$ y $SA2$. La primera parte del conjunto de entrenamiento se utilizó para llevar a cabo la tarea de aprendizaje de las funciones g_p tal y como aparecen en la Ec. 4.4, que define la función base ϕ_5. Esas funciones fueron entrenadas usando el algoritmo descrito en el capítulo anterior del presente trabajo, con las mismas secuencias de características acústicas MFCC$+\Delta+\Delta\Delta$, pero con un núcleo Gaussiano cuyos parámetros son $\sigma = 6{,}2$ y $C = 5$ para el conjunto de validación).

El segundo grupo de 100 expresiones conforma el conjunto de validación necesario para el algoritmo de alineación descrito en la *Sección* 3. Por último,

ejecutamos el algoritmo iterativo de alineamiento con las 3090 expresiones que conforman su conjunto de entrenamiento. El valor de ε en la definición de γ se configuró para ser igual a uno (es decir, 10 ms).

Como se describe en el *Apéndice A* dedicado a la descripción del corpus, el material correspondiente a las expresiones grabadas han sido subdivididas en porciones para entrenamiento y pruebas. Como parte de los datos de prueba existe un *conjunto básico* que contiene 192 frases y que constituye el material mínimo de prueba recomendado. Para el caso en que se desee realizar pruebas más extensas existe el *conjunto de pruebas completo* que consiste en un total de 1344 frases.

Evaluamos las funciones de alineamiento tanto en conjunto básico, como en el conjunto completo de pruebas del corpus TIMIT y comparamos nuestros resultados con los resultados reportados en el trabajo de Brugnara et al. [1993] y los reportados en Hosom [2002] dado que ambos trabajos son considerados una línea base en lo relativo a segmentadores de fonemas. Los resultados se presentan en la Tabla 4.1 para el caso en que usamos el corpus básico de entrenamiento y en la Tabla 4.2 para el caso en que usamos el corpus de entrenamiento completo.

Para cada valor de tolerancia $\tau \in \{10\text{ ms}, 20\text{ ms}, 30\text{ ms}, 40\text{ ms}\}$, contamos el número de predicciones cuya distancia al límite verdadero, $t = |y_i - y'_i|$, fuera menor que τ. Como se puede observar en las Tablas 4.1 y 4.2, el algoritmo discriminatorio de margen amplio presentado supera a las aproximaciones generativas basadas en MOMs descritas en Brugnara et al. [1993] para todos los valores de tolerancia predefinidos. Sin embargo es superado en 0,37 % de precisión por la propuesta presentada en Hosom [2002] para el caso en que $\mathbf{t \leq 20}$ ms. Además pudimos comprobar que los resultados obtenidos por el algoritmo presentado son los mismos que aquellos en los que se utiliza el total de las 3090 expresiones disponibles en el corpus para entrenamiento, o sólo las 50 primeras, lo que indica que el algoritmo converge después de las primeras 50 expresiones.

Usando conjunto básico de pruebas del corpus TIMIT				
	t $\leq$ **10**ms	**t** $\leq$ **20**ms	**t** $\leq$ **30**ms	**t** $\leq$ **40**ms
Discriminatorio	**78.8**	92.2	**96.1**	**98.0**
Brugnara	75.3	88.9	94.4	97.1
Hosom		**92.57**		

Tabla 4.1: Porcentaje de posicionamiento correcto de los límites de los fonemas, dada una tolerancia predefinida de **t** $\leq$ **10**ms, **t** $\leq$ **20**ms, **t** $\leq$ **30**ms y **t** $\leq$ **40**ms usando el conjunto básico de pruebas del corpus TIMIT.

Usando conjunto completo de pruebas del corpus TIMIT				
	t $\leq$ **10**ms	**t** $\leq$ **20**ms	**t** $\leq$ **30**ms	**t** $\leq$ **40**ms
Discriminatorio	**79.8**	**92.0**	**96.1**	**98.0**
Brugnara	74.6	88.8	94.1	96.8

Tabla 4.2: Porcentaje de correcto posicionamiento de los límites de los fonemas, dada una tolerancia predefinida de **t** $\leq$ **10**ms, **t** $\leq$ **20**ms, **t** $\leq$ **30**ms y **t** $\leq$ **40**ms usando el conjunto completo de pruebas del corpus TIMIT.

Capítulo 5
Reconocimiento de fonemas

Resumen:

En este capítulo se presenta un método para reconocimiento de secuencias de fonemas. El método usa un procedimiento de entrenamiento discriminatorio basado en núcleo en el que se adapta el proceso de aprendizaje con el objetivo de minimizar la distancia de Levenshtein entre la secuencia de fonemas predicha y la secuencia de fonemas correcta. El predictor de la secuencia de fonemas se diseña mapeando la expresión del habla, junto con una secuencia de fonemas tentativa a un espacio vectorial dotado de producto interno que es posible gracias a la utilización de un núcleo de Mercer. Basándonos en técnicas de margen amplio para la predicción de secuencias, es factible diseñar un algoritmo de aprendizaje que logra distinguir la secuencia de fonemas correcta de todas las demás secuencias. Se describe el algoritmo iterativo para el aprendizaje del reconocedor de secuencias de fonemas, se presentan los resultados experimentales obtenidos usando el corpus TIMIT y se comparan con los obtenidos por implementaciones de enfoques basados en MOMs. Este capítulo está organizado de la siguiente manera. En la *Sección 2* se introduce formalmente el problema del reconocimiento de secuencias de fonemas. A continuación en la *Sección 3*, se describe el método de aprendizaje propuesto. Dicho método se basa en una función no lineal de reconocimiento de fonemas usando núcleos de Mercer. En la *Sección 4* se introduce un núcleo específico para la tarea del reconocimiento de fonemas y concluimos esta presentando los resultados experimentales en la *Sección 5*.

5.1. El problema del reconocimiento de secuencias de fonemas

En el problema del reconocimiento de secuencias de fonemas, contamos con una expresión del habla y nuestro objetivo es predecir la secuencia de fonemas que le corresponde. Si representamos a una señal de voz como una secuencia de vectores de características acústicas $\overline{\mathbf{x}} = (\mathbf{x}_1, ..., \mathbf{x}_T)$, en donde $\mathbf{x}_t \in \mathcal{X}$ para toda $1 \leq t \leq T$. Cada expresión hablada corresponde a una secuencia de símbolos de fonema. De manera formal denotamos a cada símbolo de fonema como $p \in \mathcal{P}$, en donde $\mathcal{P}$ es un conjunto de estos símbolos, y denotamos a la secuencia de símbolos de fonema como $\overline{p} = (p_1, ..., p_K)$. Además denotamos como $s_k \in \mathbb{N}$, al tiempo de inicio del fonema p_k (en unidades de tramas) y denotamos como $\overline{s} = (s_1, ..., s_K)$ a la secuencia de todos los tiempos de inicio de fonema.

Naturalmente, la longitud de la señal de voz, y por lo tanto el número de fonemas, varía de una expresión a otra y por lo tanto T y K no son fijas. Denotamos como $\mathcal{P}^*$ al conjunto de todas las secuencias de longitud finita en $\mathcal{P}$ y en forma similar $\mathcal{X}$ y $\mathbb{N}^*$. Nuestro objetivo es entrenar una función f que pronostique la secuencia de fonemas correcta dada una secuencia acústica. Es decir, f es una función de $\mathcal{X}^*$ en el conjunto de secuencias de longitud finita sobre el dominio de los símbolos de fonema $\mathcal{P}^*$. También nos referimos a f como reconocedor de secuencias de fonema o predictor. El objetivo final de la predicción de la secuencia de fonemas es por lo general minimizar la distancia de Levenshtein entre la secuencia pronosticada y la correcta. La distancia de Levenshtein entre dos cadenas esta dada por el número mínimo de operaciones necesarias para transformar una cadena en otra, donde una operación puede ser una *inserción*, *eliminación* o *sustitución* de un solo carácter. La aplicación más conocida de la distancia de Levenshtein se puede encontrar en aplicaciones tales como correctores ortográficos, sin embargo, también se utiliza para evaluar la calidad de la predicción de una secuencia de fonemas [Lee y Hon, 1989].

A lo largo de esta parte del trabajo denotremos como $\gamma(\overline{p}, \overline{p}')$ a la distancia de Levenshtein entre la secuencia de fonemas pronosticada $\overline{p}'$ y la secuencia de fonemas verdadera $\overline{p}$. En la siguiente sección se presenta un algoritmo que tiene como objetivo minimizar la distancia de Levenshtein entre la secuencia de fonemas pronosticada y la secuencia de fonemas correcta.

5.2. El algoritmo de aprendizaje

En esta sección se describe un algoritmo de aprendizaje supervisado discriminatorio para el aprendizaje de una secuencia de fonemas f a partir de un conjunto de ejemplos de entrenamiento. Cada ejemplo del conjunto de entrenamiento se compone de una señal acústica $\overline{\mathbf{x}}$, una secuencia de fonemas $\overline{p}$ y una secuencia de tiempos de inicio de fonemas $\overline{s}$. La construcción del algoritmo se basa en una función característica predefinida $\phi : \mathcal{X}^* \times (\mathcal{P} \times \mathbb{N})^* \rightarrow \mathcal{H}$, donde $\mathcal{H}$ es un *espacio de Hilbert generado por núcleo*[1]. La entrada de esta función es una representación acústica $\overline{\mathbf{x}}$, junto con una secuencia candidata o tentativa $\overline{p}$ de símbolos de fonema y una secuencia candidata $\overline{s}$ de tiempos de inicio de fonema. La función característica devuelve un vector en $\mathcal{H}$. De manera intuitiva vemos que cada elemento del vector representa un valor de confianza en la secuencia de fonemas sugerida. Por ejemplo, un elemento de la función característica puede sumar el número de veces que el fonema p viene después del fonema p', mientras que otro elemento de dicha función extrae las propiedades de cada vector de características acústicas $\mathbf{x}_t$, siempre que el fonema p haya sido pronunciado en el tiempo t. La descripción concreta y más detallada de la forma de la función característica se pospone para la *Sección 4*.

Nuestro objetivo es entrenar un reconocedor de secuencias de fonemas f, que tome como entrada una secuencia de características acústicas $\overline{\mathbf{x}}$ y devuelva una secuencia de símbolos de fonema $\overline{p}$. La forma de la función f que usamos es

$$f(x) = \underset{\overline{p}}{\operatorname{argmax}}(\underset{\overline{s}}{\operatorname{máx}}\, \mathbf{w} \cdot \phi(\overline{\mathbf{x}}, \overline{p}, \overline{s})), \tag{5.1}$$

en donde $\mathbf{w} \in \mathcal{H}$ es un vector de pesos de importancia que debe aprenderse. Es decir, f devuelve una sugerencia para una secuencia de fonemas mediante la maximización de la suma ponderada de las puntuaciones devueltas por los elementos de la función característica. Aprender el vector de pesos $\mathbf{w}$ es similar a la estimar los parámetros de las funciones de densidad de probabilidad local en los enfoques basados en MOMs. El enfoque que presentamos aquí sin embargo, no requiere que $\mathbf{w}$ adopte una forma probabilística. La maximización definida por la Ec. 5.1 tiene un número exponencialmente grande de posibles secuencias de fonemas. Sin embargo, al igual que en el enfoque basa-

[1]Mejor conocido como RKHS del inglés Reproducing Kernel Hilbert Space.

do en MOMs, si la función característica ϕ se puede descomponer, entonces la optimización de la Ec. 5.1 puede calcularse eficientemente utilizando un procedimiento de programación dinámica.

A continuación describimos un algoritmo iterativo simple para el aprendizaje del vector de pesos $\mathbf{w}$. El algoritmo recibe como entrada un conjunto $S = \{(\overline{\mathbf{x}}_1, \overline{p}_1, \overline{s}_1), ..., (\overline{\mathbf{x}}_m, \overline{p}_m, \overline{s}_m)\}$ de ejemplos de entrenamiento, en donde inicialmente establecemos $\mathbf{w} = 0$. En cada iteración el algoritmo actualiza el valor de $\mathbf{w}$ según el $i-$ésimo ejemplo en S, como se describe a más adelante. Denotamos como $\mathbf{w}$ al valor del vector de pesos antes de la $i-$ésima iteración. Sea $(\overline{p}'_i, \overline{s}'_i)$ la secuencia de fonemas predicha para el $i-$ésimo ejemplo de acuerdo con $\mathbf{w}_{i-1}$,

$$(\overline{p}'_i, \overline{s}'_i) = \underset{(\overline{p},\overline{s})}{\operatorname{argmax}} \mathbf{w}_{i-1} \cdot \phi(\overline{\mathbf{x}}_i, \overline{p}, \overline{s}). \tag{5.2}$$

Establecemos el vector de pesos $\mathbf{w}_i$ para que sea el minimizador del siguiente problema de optimización,

$$\underset{\mathbf{w}\in\mathcal{H},\xi\geq 0}{\text{mín}} \quad \frac{1}{2}\left\|\mathbf{w} - \mathbf{w}_{i-1}\right\|^2 + C\xi, \tag{5.3}$$

tal que $\mathbf{w}\cdot\phi(\overline{\mathbf{x}}_i, \overline{p}_i, \overline{s}_i) - \mathbf{w}\cdot\phi(\overline{\mathbf{x}}_i, \overline{p}'_i, \overline{s}'_i) \geq (\overline{p}_i, \overline{p}'_i) - \xi$, en donde C se utiliza como parámetro para ajuste del equilibrio entre complejidad y precisión (véase Cristianini y Shawe-Taylor [2000]) y ξ es una variable de holgura no negativa, que indica la pérdida en el $i-$ésimo ejemplo. Buscamos minimizar la pérdida en el ejemplo actual, es decir, la variable de holgura, mientras mantenemos el vector de pesos $\mathbf{w}$ lo más cerca posible al vector de pesos anterior $\mathbf{w}_{i-1}$. La restricción provoca que la proyección de la secuencia correcta de fonemas $(\overline{\mathbf{x}}_i, \overline{p}_i, \overline{s}_i)$ en $\mathbf{w}$ sea de dimensión superior a la proyección de la secuencia de fonemas predicha $(\overline{p}'_i, \overline{s}'_i)$ en $\mathbf{w}$, en por lo menos la distancia de Levenshtein existente entre ellos. Utilizando una deducción similar a la presentada en el capítulo anterior, se puede demostrar que la solución al problema de optimización enunciado es

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \alpha_i \Delta\phi_i, \tag{5.4}$$

en donde $\Delta\phi_i = \phi(\overline{\mathbf{x}}_i, \overline{p}_i, \overline{s}_i) - \phi(\overline{\mathbf{x}}_i, \overline{p}'_i, \overline{s}'_i)$. El valor del escalar α_i se basa en la distancia de Levenshtein $\gamma(\overline{p}_i, \overline{p}'_i)$, las diferentes puntuaciones que $\overline{p}_i$ y $\overline{p}'_i$ reciben son de acuerdo a $\mathbf{w}_{i-1}$, y a un parámetro C.

Formalmente, decimos que,

$$\alpha_i = \text{mín} \left\{ C, \frac{\text{máx}\{\gamma(\overline{p}_i, \overline{p}'_i) - \mathbf{w}_{i-1} \cdot \Delta\phi_i, 0\}}{\|\Delta\phi_i\|^2} \right\}. \tag{5.5}$$

El pseudocódigo del Algoritmo 2 en línea para clasificación de fonemas se muestra más adelante. Para concluir esta sección, ampliamos la noción de *familia de reconocedores de secuencias de fonemas* expresada en la Ec. 5.1 a las funciones de reconocimiento no lineal. Esta extensión se basa en los núcleos de Mercer usados con frecuencia en los algoritmos de MVS (véase Vapnik [1998]). Hay que recordar que la regla actualizada del algoritmo es $\mathbf{w}_i = \mathbf{w}_{i-1} + \alpha_i \Delta\phi_i$ y que el vector de pesos inicial es $\mathbf{w}_0 = 0$. Por lo tanto $\mathbf{w}_i$ se puede reescribir como $\mathbf{w}_i = \sum_{\mathbf{j=1}}^{\mathbf{i}} \alpha_j \Delta\phi_j$ y f puede reescribirse como

$$f(\overline{\mathbf{x}}) = \underset{\overline{p}}{\text{argmax}}\, \underset{\overline{s}}{\text{máx}} \sum_{j=1}^{i} \alpha_j \left(\Delta\phi_j \cdot \phi(\overline{\mathbf{x}}, \overline{p}, \overline{s}) \right). \tag{5.6}$$

Sustituyendo la definición de $\Delta\phi_j$ y reemplazando el producto interno en la Ec. 5.6 con el operador del núcleo $\mathcal{K}(\cdot, \cdot)$ que satisface las condiciones de Mercer (según se explica en Vapnik [1998]), obtenemos la siguiente *función no lineal* para reconocimiento de fonemas

$$f(\overline{\mathbf{x}}) = \underset{\overline{p}}{\text{argmax}}\, \underset{\overline{s}}{\text{máx}} \sum_{j=1}^{i} \alpha_j \left(\mathcal{K}(\overline{\mathbf{x}}_j, \overline{p}_j, \overline{s}_j; \overline{\mathbf{x}}, \overline{p}, \overline{s}) - \mathcal{K}(\overline{\mathbf{x}}_j, \overline{p}'_j, \overline{s}'_j; \overline{\mathbf{x}}, \overline{p}, \overline{s}) \right). \tag{5.7}$$

Se puede verificar que la definición de α_i expresada en la Ec. 5.5 también puede expresarse usando el operador del núcleo.

5.3. Función característica para reconocimiento de secuencias de fonemas

En esta sección se describe la función característica utilizada. Como se mencionó en la sección anterior, nuestro objetivo es la implementación de un

Algoritmo 2 Algoritmo iterativo para reconocimiento de fonemas

Entrada: Conjunto de datos de entrenamiento $S = \{(\overline{\mathbf{x}}_i, \overline{p}_i, \overline{s}_i)\}_{i=1}^{m}$.

Conjunto de datos de validación $S_{validación} = \{(\overline{\mathbf{x}}_i, \overline{p}_i, \overline{s}_i)\}_{i=1}^{m_{validación}}$.

Parámetro C.

Salida: El vector de pesos $\mathbf{w}^*$ que alcanza el más alto desempeño con el conjunto de datos de validación $S_{validación}$:

$$\mathbf{w}^* = \underset{\mathbf{w}\in\{\mathbf{w_1},\ldots,\mathbf{w_m}\}}{\operatorname{argmin}} \sum_{j=1}^{m_{validación}} \gamma(\overline{p}_j, f(\overline{\mathbf{x}}_j))$$

Inicializar: $\mathbf{w}_0 = 0$

Para $i = 1, 2, ..., m$ **hacer**

Predecir $(\overline{p}'_i, \overline{s}'_i) = \underset{\overline{p},\overline{s}}{\operatorname{argmax}}\, \mathbf{w}_{i-1} \cdot \phi(\overline{\mathbf{x}}_i, \overline{p}, \overline{s})$.

Hacer $\Delta\phi_i = \phi(\overline{\mathbf{x}}_i, \overline{p}_i, \overline{s}_i) - \phi(\overline{\mathbf{x}}_i, \overline{p}'_i, \overline{s}'_i.)$

Hacer $\ell(\mathbf{w}_{i-1}; \overline{\mathbf{x}}_i, \overline{p}_i, \overline{s}_i) = \text{máx}\{\gamma(\overline{p}_i, \overline{p}'_i) - \mathbf{w}_{i-1} \cdot \Delta\phi_i, 0\}$.

Si $\ell(\mathbf{w}_{i-1}; \overline{\mathbf{x}}_i, \overline{p}_i, \overline{s}_i) > 0$ **entonces**

Hacer $\alpha_i = \text{mín}\left\{C, \dfrac{\ell(\mathbf{w}_{i-1}; \overline{\mathbf{x}}_i, \overline{p}_i, \overline{s}_i)}{\|\Delta\phi_i\|^2}\right\}$

Actualizar: $\mathbf{w}_i = \mathbf{w}_{i-1} + \alpha_i \cdot \Delta\phi_i$

fin Si
fin Para

reconocedor de secuencias de fonemas no lineal utilizando núcleos de Mercer. Para ello en lugar de describir la función característica, describimos en primer lugar al operador del núcleo que calcula implícitamente el producto interno $\phi(\overline{\mathbf{x}}, \overline{p}, \overline{s}) \cdot \phi(\overline{\mathbf{x}}', \overline{p}', \overline{s}')$. Para simplificar la notación denotamos como $\overline{\mathbf{z}}$ a la tripleta $(\overline{\mathbf{x}}, \overline{p}, \overline{s})$ y de manera similar $\overline{\mathbf{z}}'$ denota a $(\overline{\mathbf{x}}', \overline{p}', \overline{s}')$. Los operadores del núcleo $\mathcal{K}(\overline{\mathbf{z}}, \overline{\mathbf{z}}')$ que presentamos, se pueden escribir como una suma ponderada de tres operadores del núcleo

$$\mathcal{K}(\overline{\mathbf{z}}, \overline{\mathbf{z}}') = \sum_{i=1}^{3} \beta_i \mathcal{K}_i(\overline{\mathbf{z}}, \overline{\mathbf{z}}'),$$

en donde β_i son parámetros positivos. A continuación describimos los tres operadores del núcleo utilizados. El primer operador del núcleo $\mathcal{K}_1$, es similar al del modelo acústico, que aparece en los MOMs. Decimos en primer lugar que para cada fonema $p \in \mathcal{P}$ sea $T_p(\overline{\mathbf{z}})$ el conjunto de todos los tiempos (de la trama) en los que el fonema p es pronunciado. Es decir, $T_p(\overline{\mathbf{z}}) = \{t : \exists k, p_k = p \wedge s_k \leq t \leq s_{k+1}\}$. Usando esta definición, el primer operador del núcleo se define como

$$\mathcal{K}_1(\overline{\mathbf{z}}, \overline{\mathbf{z}}') = \sum_{p \in \mathcal{P}} \sum_{t \in T_p(\overline{\mathbf{z}})} \sum_{\tau \in T_p(\overline{\mathbf{z}})} \exp\left(-\frac{\|\mathbf{x}_t - \mathbf{x}'_\tau\|^2}{2\sigma^2} \right),$$

en donde σ es una constante predefinida. Es importante señalar que el término $\mathcal{K}_1(\overline{\mathbf{z}}_i, \overline{\mathbf{z}}) - \mathcal{K}_1(\overline{\mathbf{z}}'_i, \overline{\mathbf{z}}')$ en la Ec. 5.7 puede escribirse de forma más compacta como sigue

$$\sum_{p \in \mathcal{P}} \left[\sum_{t \in T_p(\overline{\mathbf{z}}_i)} \sum_{\tau \in T_p(\overline{\mathbf{z}})} \exp\left(-\frac{\|\mathbf{x}_{i,t} - \mathbf{x}_\tau\|^2}{2\sigma^2} \right) - \sum_{t \in T_p(\overline{\mathbf{z}}'_i)} \sum_{\tau \in T_p(\overline{\mathbf{z}})} \exp\left(-\frac{\|\mathbf{x}_{i,t} - \mathbf{x}_\tau\|^2}{2\sigma^2} \right) \right]$$

Entonces reescribiendo el término dentro de los corchetes de forma compacta, tenemos que

$$\sum_{p \in \mathcal{P}} \sum_{t \in T_p(\overline{\mathbf{z}})} \sum_{t=1}^{|\overline{\mathbf{x}}_i|} \psi(t; \overline{\mathbf{z}}_i, \overline{\mathbf{z}}'_i) \exp\left(-\frac{\|\mathbf{x}_{i,t} - \mathbf{x}_\tau\|^2}{2\sigma^2} \right), \tag{5.8}$$

en donde

$$\psi\left(t;\overline{\mathbf{z}}_i,\overline{\mathbf{z}}'_i\right)=\begin{cases}1 & t\in T_p(\overline{\mathbf{z}}_i)\wedge t\notin T_p(\overline{\mathbf{z}}'_i)\\ -1 & t\notin T_p(\overline{\mathbf{z}}_i)\wedge t\in T_p(\overline{\mathbf{z}}'_i)\\ 0 & \text{de cualquier otra manera}\end{cases}$$

En particular para cada trama $\mathbf{x}_{i,t}$ en donde $\overline{\mathbf{z}}_i$ y $\overline{\mathbf{z}}'_i$ coincidan respecto del fonema pronunciado, el valor de $\psi\left(t;\overline{\mathbf{z}}_i,\overline{\mathbf{z}}'_i\right)$ es cero, lo que significa que en la trama $\mathbf{x}_{i,t}$ no se realiza la predicción.

El segundo operador del núcleo $\mathcal{K}_2$ es similar a un modelo de duración de fonema, por lo tanto es ajeno a la señal de voz en sí y sólo tiene por objeto analizar la duración de cada fonema. Sea $\mathcal{D}$ el conjunto de umbrales predefinidos. Para cada $p\in\mathcal{P}$ y $d\in\mathcal{D}$ sea $N_{p,d}(\overline{\mathbf{z}})$ el número de tiempos en que el fonema p aparece en $\overline{p}$, mientras que su duración fue de por lo menos d, es decir $N_{p,d}(\overline{\mathbf{z}})=|\{k:p_k=p\wedge(s_{k+1}-s_k)\geq d\}|$. Usando esta notación, $\mathcal{K}_2$ se puede definir como

$$\mathcal{K}_2\left(\overline{\mathbf{z}},\overline{\mathbf{z}}'\right)=\sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}N_{p,d}(\overline{\mathbf{z}})N_{p,d}(\overline{\mathbf{z}}').$$

El último de los operadores de núcleo $\mathcal{K}_3$ es similar al modelo de transición de fonema. Sea $A(p',p)$ una estimación de la matriz de probabilidad de transición del fonema p' al fonema p. Además, sea Θ un conjunto de valores de umbral. Para cada $\theta\in\Theta$ sea $N_\theta(\overline{\mathbf{z}})$ el número de veces que cambiamos del fonema p_{k-1} al fonema p_k tal que $A(p_{k-1},p_k)$ sea por lo menos el valor de θ, es decir que $N_\theta(\overline{\mathbf{z}})=|\{k:A(p_{k-1},p_k)\geq\theta\}|$. Usando esta notación, $\mathcal{K}_3$, se define como

$$\mathcal{K}_3\left(\overline{\mathbf{z}},\overline{\mathbf{z}}'\right)=\sum_{\theta\in\Theta}N_\theta(\overline{\mathbf{z}})N_\theta(\overline{\mathbf{z}}').$$

Hay que recordar que el cálculo de f requiere resolver el siguiente problema de optimización

$$f(\overline{\mathbf{x}})=\underset{\overline{p}}{\operatorname{argmax}}\,\underset{\overline{s}}{\operatorname{máx}}\sum_{j=1}^{m}\alpha_j\left(\mathcal{K}\left(\overline{\mathbf{z}}_j,\overline{\mathbf{z}}\right)-\mathcal{K}\left(\overline{\mathbf{z}}'_i,\overline{\mathbf{z}}\right)\right).$$

Una búsqueda directa del maximizador no es factible ya que el número de secuencias de fonemas posibles es exponencial. Afortunadamente, el operador del núcleo que presentamos puede descomponerse y por lo tanto se puede encontrar la mejor secuencia de fonemas en tiempo polinomial con programación dinámica (de manera similar al procedimiento de Viterbi que a menudo se implementa en el enfoque basado en MOMs [Rabiner y Juang, 1993]).

5.4. Resultados experimentales

Para validar la eficacia del método presentado se realizaron experimentos con el corpus TIMIT. Todos los experimentos descritos aquí siguen la misma metodología. Hemos dividido la porción de entrenamiento del corpus TIMIT (excluyendo las expresiones habladas en SA1 y SA2) en dos partes disjuntas que contienen 3600 y 96 expresiones. La primer parte se utilizó como conjunto de entrenamiento y la segunda parte se utilizó como conjunto de validación. Se extrajeron los coeficientes cepstrales en las frecuencias Mel[2], de la expresión hablada junto con su primera y segunda derivada en forma estándar, con *Substracción de la Media del Cepstrum*[3]. Se eliminaron los silencios iniciales y finales de cada expresión. El conjunto original de 61 fonemas en el corpus TIMIT fue mapeado al conjunto de 39 fonemas como se propone en Lee y Hon [1989]. El rendimiento fue evaluado en el conjunto básico de entrenamiento del corpus TIMIT mediante el cálculo de la distancia de Levenshtein entre la secuencia de fonemas pronosticada y la correcta.

Aplicamos el método como se discutió en las *Secciones* 3 *y* 4 con $\sigma^2 = 6$, $C = 80$, $\mathcal{D} = \{5, 10, 15, ..., 40\}$, $\beta = \{1, 0{,}02, 0{,}005\}$ y $\Theta = \{0{,}1, 0{,}2, ..., 0{,}8, 0{,}9\}$. Se compararon los resultados al aplicar el método descrito en el presente capítulo, con los resultados descritos en el trabajo de Lee y Hon [1989] cuyos resultados usamos como línea base y punto de comparación en esté y anteriores capítulos y que se basa en una implementación del enfoque basado en MOMs. Donde a cada fonema se le representa con un MOM de 5 estados emitidos en una topología de izquierda a derecha y con 40 Gaussianas de

[2]Mejor conocidos como MFCC del inglés Mel Frequency Cepstral Coefficients.
[3]Mejor conocido como CMS, del inglés Cepstral Mean Subtraction.

matriz de covarianza diagonal. En donde según se describe todos los hiperparámetros incluyendo el número de estados y de Gaussianas por estado, fueron ajustados con base al conjunto de validación.

Los resultados globales se presentan en las Tablas 5.1 y 5.2. Se reporta el número de inserciones, eliminaciones y substituciones, según fueron calculadas por la distancia de Levenshtein. Se define a la distancia de Levenshtein como la suma de las inserciones, eliminaciones y substituciones y establecemos como *precisión* al 100 % menos la distancia de Levenshtein y como *correcto* a la precisión mas las inserciones. Como se puede observar, el método basado en MOM supera en términos de precisión al método propuesto basado en núcleo, debido principalmente al alto nivel de inserciones de nuestro método, lo que sugiere que debemos explorar la creación de un mejor modelo.

	Inserciones	**Eliminaciones**	**Susutituciones**
Núcleo	11.12	10.2	27.4
MOM	**10.79**	**9.72**	**26.22**

Tabla 5.1: Comparación de los resultados del reconocimiento de fonemas obtenidos por el algoritmo discriminatorio basado en núcleo y la aproximación basada en Modelos Ocultos de Markov según se reporta en el trabajo de Lee y Hon [1989]. Porcentaje de inserciones, eliminaciones y sustituciones.

	Precisión	**Reconocimiento correcto**
Núcleo	51.2	62.4
MOM	**53.27**	**64.07**

Tabla 5.2: Comparación de los resultados del reconocimiento de fonemas obtenidos por el algoritmo discriminatorio basado en núcleo y la aproximación basada en Modelos Ocultos de Markov según se reporta en el trabajo de Lee y Hon [1989]. Porcentajes de precisión y de reconocimiento correcto.

A pesar de ello consideramos que el potencial del método presentado es más grande de lo que reflejan los resultados hasta ahora obtenidos y en las secciones dedicadas a conclusiones y trabajo a futuro se discuten algunas posibles mejoras. Consideramos también que si bien ha habido un esfuerzo contínuo y amplio en el uso de los enfoques basados en MOMs para el reconocimiento de secuencias de fonemas, el método presentado aquí es innovador y por esto es que la elección de las posibles configuraciones de características y operadores del núcleo aun no han sido estudiadas de manera exhaustiva, así como tampoco los resultados de dichas elecciones.

Capítulo 6
Reconocimiento de palabras clave

Resumen:

En este capítulo se presenta un método para la detección y localización de palabras clave en expresiones habladas. Dada una palabra representada como una secuencia de fonemas y una expresión hablada, el detector de palabras clave predice el intervalo de tiempo en que ocurre dicha secuencia de fonemas dentro de la expresión, junto con un nivel de confianza de dicha ocurrencia. Tomando como fundamento las técnicas usadas por métodos de margen amplio y de núcleo para la predicción de secuencias completas, se formula el problema del entrenamiento del detector de palabras clave como una tarea discriminatoria, en donde los parámetros del modelo son elegidos de forma que la expresión hablada en la que se pronuncia la palabra clave, tiene un mayor nivel de confianza que en cualquier otra expresión. A diferencia de otros enfoques, el método propuesto emplea un procedimiento de aprendizaje discriminatorio, que en la fase de aprendizaje tiene como objetivo maximizar el área bajo la curva ROC, ya que esta es la medida más común para evaluar el desempeño de los detectores de palabras clave. Se muestra teórica y empíricamente que del método de entrenamiento propuesto resulta una gran *área bajo la curva* ROC. Se presenta un algoritmo iterativo para entrenar al reconocedor de palabras clave de manera eficiente. El enfoque propuesto contrasta con la estrategia de reconocimiento basada en modelos ocultos de Markov. Se presentan los resultados experimentales obtenidos, que muestran las ventajas del uso del enfoque presentado sobre las alternativas basadas en MOMs.

6.1. Introducción

El reconocimiento de palabras clave tiene por objeto detectar (o reconocer) cualquier palabra relevante para un usuario o para un sistema, en una determinada señal de voz, perteneciente a una determinada expresión hablada o frase pronunciada. Esta tarea es importante para numerosas aplicaciones, como la recuperación de correos de voz, detección de comandos de voz y la detección y recuperación de términos hablados, por mencionar solo algunas. El trabajo previo se ha centrado principalmente en diversas variantes de los MOMs para enfrentar este problema intrínsecamente secuencial (véanse por ejemplo Benayed et al. [2003] y Szöke et al. [2005]). Mientras que los enfoques basados en MOMs constituyen el estado del arte, sufren de varias conocidas limitaciones. La mayoría de estas limitaciones no son específicas del problema de reconocimiento de palabras clave sino que son comunes a otras tareas como el reconocimiento automático del habla, como se señaló en la sección sobre las *limitaciones y problemas potenciales del enfoque probabilístico*, en el *Capítulo* 2 sobre antecedentes del presente trabajo.

Otras deficiencias son específicas de la aplicación de los MOMs a la tarea del reconocimiento de palabras clave. En particular, la escasa presencia de ciertas palabras en el corpus de entrenamiento a menudo requiere modificaciones *ad-hoc* de la topología del MOM, de las probabilidades de transición o del algoritmo de decodificación. La limitación más grave de los enfoques basados en MOMs la encontramos en el objetivo de su entrenamiento. Típicamente, el entrenamiento en los MOMs no tiene por objeto minimizar la pérdida directamente relacionada con el rendimiento del reconocedor, es decir tiene como objetivo maximizar la probabilidad de las expresiones transcritas, y no proporciona ninguna garantía en términos del rendimiento en la detección de palabras clave.

El rendimiento de un sistema de detección de palabras clave a menudo se mide con la *curva de características operativas del receptor* (mejor conocida como curva ROC[1]), es decir, mediante la representación gráfica de la tasa de

[1]Por sus siglas en inglés Receiver Operating Characteristic. Se introdujo originalmente para la teoría de detección de señales, en relación con el estudio de las señales de radio, y se ha utilizado desde entonces en muchas otras aplicaciones, en especial para toma de decisiones. En años recientes, han generado un mayor interés para las investigaciones en minería de datos y aprendizaje automático, para la evaluación y selección de mejores

verdaderos positivos (identificación correcta de una palabra clave) en función de la tasa de falsos positivos (error en la identificación de una palabra clave). Cada punto de la curva ROC representa el rendimiento del sistema para un determinado ajuste del balance entre el logro de una alta tasa de verdaderos positivos y una baja tasa de falsos positivos [Cortes y Mohri, 2004]. Dado que la configuración de dicho balance no siempre está definida de antemano, los sistemas normalmente se evalúan comparando el rendimiento promedio en todos los puntos de operación, prefiriéndose aquellos sistemas que alcanzan el valor más grande del *área bajo la curva ROC* (ABC)[2].

Presentamos un enfoque discriminatorio basado en métodos de margen amplio, para el proceso de aprendizaje necesario en la detección de una palabra clave. Es decir, se propone un enfoque alternativo para resolver el problema del reconocimiento de palabras clave que se basa en trabajos recientes sobre métodos discriminatorios de margen amplio y de núcleo, como un intento por superar algunos de los problemas inherentes de los métodos basados en MOMs. Dicho enfoque resuelve directamente el problema de detección de palabras clave en lugar de utilizar un reconocedor del habla continua para vocabulario grande (como en Szöke et al. [2005]), y no necesita de un modelo para basura o de relleno (como en Silaghi et al. [1999]).

Una de las ventajas de los métodos discriminatorios de margen amplio y de núcleo proviene del hecho de que la función objetivo utilizada durante la fase de aprendizaje está estrechamente unida a la tarea de decisión que se necesita llevar a cabo. El método presentado se basa en los recientes avances en la investigación de clasificadores de secuencias con máquinas núcleo (véanse Shalev-Shwartz et al. [2004] y Taskar et al. [2003]). El detector de palabras clave se basa en el mapeo de la señal de voz junto con la palabra clave objetivo a un espacio vectorial dotado de un producto interno. Es en este aspecto que este enfoque esta relacionado con las máquinas de vectores de soporte (MVS), las cuales ya han sido implementadas con éxito en aplicaciones de procesamiento de voz (véanse Salomon et al. [2002] y Keshet et al. [2001]).

---

modelos.

[2]En el *Apéndice B* se revisa con detalle la construcción y utilidad de dicha representación gráfica, así como la obtención del área bajo la curva ROC, para el detector de palabras clave presentado.

Sin embargo, el modelo que se presenta en este documento difiere significativamente de las aproximaciones clásicas con MVS debido a la naturaleza secuencial del problema de detección de palabras clave.

La función de detección de palabras clave tiene como entrada una secuencia de fonemas que representa a la palabra clave y además una expresión hablada. Como salida una predicción del intervalo de tiempo en que ocurre la palabra clave en la expresión hablada, así como un valor de confianza respecto de dicha predicción. El objetivo del algoritmo de entrenamiento es el de maximizar el área bajo la curva (ABC) usando el conjunto de datos de entrenamiento y el de prueba (ocultos). A una frase en el conjunto de entrenamiento en la que se pronuncia la palabra clave la llamaremos *frase positiva*, y por lo tanto, una frase en la que la palabra clave no se pronuncia la conoceremos como una *frase negativa*. Entonces se formula la tarea de entrenamiento como un problema de estimación de los parámetros del modelo de tal manera que el valor de confianza del lapso de tiempo correcto en una frase positiva sea mayor que el valor de confianza en cualquier intervalo de tiempo de cualquier frase negativa.

Formalmente este problema se plantea como un *problema de optimización convexa con restricciones*. La solución a este problema de optimización es una función que analíticamente demuestra que puede alcanzar altos valores de ABC con el conjunto de datos de entrenamiento y que probablemente también tiene buenas propiedades de generalización en los datos de prueba. Por otra parte, en comparación con los MOMs, el enfoque presentado se basa en un procedimiento de optimización convexa, que converge a óptimos globales y que no está basado en un marco probabilístico, que ofrece además una mayor flexibilidad en la selección de la importancia relativa de los modelos de duración con respecto al modelado acústico.

El resto de este capítulo está organizado de la siguiente manera, en la *Sección 2* se describe brevemente el estado del arte y el trabajo previo en la detección de palabras clave, en la *Sección 3* de define el problema del reconocimiento de palabras clave, en la *Sección 4* se presenta a la función de pérdida y parametrización del modelo, en la *Sección 5* se presenta el algoritmo de entrenamiento iterativo y por último en la *Sección 6* se presentan los resultados obtenidos en diferentes experimentos, comparando el modelo propuesto con una solución basada en MOMs.

6.2. Estado del arte y trabajo previo

La investigación en la detección de palabras clave se ha llevado a cabo paralelamente al desarrollo del área del reconocimiento automático del habla (RAH). Al igual que en el RAH, la detección de palabras clave ha sido abordada inicialmente usando modelos basados en *alineamiento temporal dinámico* (comúnmente conocido como DTW[3]) [Higgins y Wohlford, 1985]. Posteriormente se introdujeron los enfoques basados en MOMs discretos [Kawabata et al., 1988] y finalmente, los MOMs discretos fueron reemplazados por MOMs contínuos [Rabiner y Juang, 1993]. El objetivo central de un sistema de detección de palabras clave es distinguir entre expresiones en las que se pronuncia la palabra clave y expresiones en las que la palabra clave no es pronunciada.

Para ello, las primeras aproximaciones basadas en DTW proponían calcular la distancia de alineamiento entre la plantilla de una expresión que representa la palabra clave objetivo y todos los segmentos posibles de la señal de prueba [Bridle, 1973]. En este contexto, la palabra clave se considera detectada en un determinado segmento de la expresión de prueba siempre que la distancia de alineamiento sea inferior a un umbral predefinido. Estas aproximaciones se ven sin embargo gravemente afectadas por el cambio de hablante (falta de coincidencia) y las diferentes condiciones de grabación entre la secuencia de la plantilla y la señal de prueba.

Posteriormente y para ganar robustez se propuso calcular las distancias de alineamiento no sólo con respecto a la plantilla de la palabra clave objetivo, sino también con respecto a otras plantillas de palabras clave [Higgins y Wohlford, 1985]. Es decir, dada una expresión hablada del conjunto de datos de prueba, el sistema identifica la concatenación de aquellas plantillas con la menor distancia a la señal y la palabra clave es considerada como detectada si esta concatenación contiene a la plantilla de la palabra clave objetivo. Por lo tanto, la distancia de alineamiento de la palabra clave no se considera como una cifra absoluta, sino relativa a las distancias a otras plantillas, lo que aumenta la robustez con respecto a los cambios en las condiciones de grabación.

Junto con el desarrollo en la investigación de las tecnologías del habla, se

[3]Por sus siglas en inglés Dynamic Time Warping.

recolectaron cantidades de datos de expresiones del habla (con su respectivo etiquetado) cada vez más grandes y las técnicas basadas en DTW comenzaron a mostrar su incapacidad para aprovechar estos grandes conjuntos de datos de entrenamiento. En particular, los grandes corpus que contienen miles de palabras y decenas de casos por cada palabra (diferentes realizaciones de la misma frase y palabra). Si se toma a cada realización como una plantilla, la concatenación de búsquedas por la mejor plantilla se convierte en una tarea prohibitivamente cara.

Como consecuencia, se introdujo el uso de MOMs discretos para RAH [Bahl et al., 1986] y luego para la detección de palabras clave [Kawabata et al., 1988; Wilpon et al., 1990]. Los MOMs discretos presuponen que los vectores de características acústicas cuantizados, que representan a la expresión de entrada, son independientes de las variables de estado oculto. Este tipo de modelo presenta varias ventajas en comparación con los enfoques basados en DTW, incluyendo una mayor robustez a los cambios de hablante y condiciones de grabación, sobre todo cuando se dispone de varias expresiones para entrenamiento de la palabra clave objetivo.

Sin embargo, la evolución más importante introducida junto con los MOMs se encuentra sin duda en el desarrollo de modelos basados en fonos y tri-fonos [Kawabata et al., 1988; Lee y Hon, 1989], en los que un modelo de palabra se compone de varios modelos de sub-unidad que se pueden compartir entre palabras. Esto significa que el modelo de una palabra dada no sólo se beneficia de las expresiones de entrenamiento que contienen esta palabra, sino también de todas las expresiones que contiene sus sub-unidades. Además, con este enfoque se logra una gran escalabilidad ya que el número de parámetros acústicos es proporcional a la cantidad de sub-unidades y no crece linealmente con el tamaño del corpus, a diferencia de los enfoques basados en plantillas. Una ventaja adicional de modelado basado en fonos es la capacidad de detectar palabras no disponibles en tiempo de entrenamiento, ya que este paradigma permite construir un nuevo modelo de palabra aprovechando la composición de modelos de sub-unidad ya entrenados. Este aspecto es muy importante, ya que en la mayoría de las aplicaciones el conjunto de palabras clave para prueba no se conoce de antemano.

Poco después de la aplicación de los MOMs discretos a problemas del habla, se introdujeron en el área del RAH los MOMs de de densidad continua [Rabiner y Juang, 1993]. Los MOMs continuos eliminan la necesidad de

la cuantización vectorial acústica, debido a que las distribuciones asociadas con los estados del modelo son densidades continuas, a menudo modeladas mediante modelos de mezclas de Gaussianas (MMG). El aprendizaje de los parámetros de los MMG y de las probabilidades de transición entre estados se lleva a cabo en un marco único e integrado, lo que maximiza la probabilidad de los datos de entrenamiento dada su transcripción a través del algoritmo de maximización de la esperanza (ME). Este enfoque ha demostrado ser más eficaz y permite mayor flexibilidad para la adaptación del hablante o del canal [Rabiner y Juang, 1993]. Ahora mismo este es el método más utilizado tanto para reconocimiento automático del habla, como para detección de palabras clave.

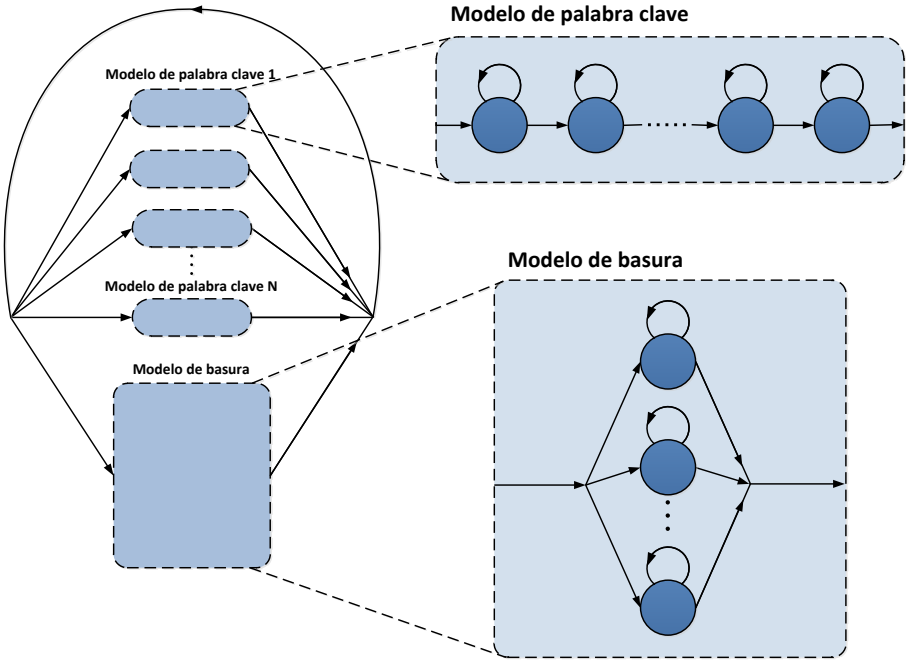

Figura 6.1: Topología básica del modelo oculto de Markov para la detección de palabras clave. En este enfoque se verifica si la mejor ruta, según el algoritmo de Viterbi, pasa a través del sub modelo de palabras clave.

En el contexto de la detección de palabras clave, se han propuesto diferen-

tes estrategias basadas en MOMs continuos. En la mayoría de los casos, un MOM basado en sub-unidades es entrenado con un corpus grande de datos transcritos de expresiones habladas y el nuevo modelo se va creando a partir de los modelos de sub-unidades. Este modelo se compone de dos partes, un *MOM de palabras clave* y un *MOM de basura* (o de relleno), que respectivamente modelan las partes de la señal de las palabras clave y de aquellas que no lo son. Dicha topología se muestra en la Fig. 6.1.

Teniendo en cuenta este modelo, la detección de palabras clave se realiza mediante la búsqueda de la secuencia de estados que produce la mayor probabilidad para la secuencia de prueba, obtenida a través de la decodificación Viterbi. La detección de palabras clave se determina comprobando si la mejor ruta de Viterbi pasa a través del modelo de la palabra clave o no. En tal modelo la selección de las probabilidades de transición en la palabra clave configura el equilibrio entre obtener una tasa baja de falsas alarmas (detección de una palabra clave cuando no esta presente) y una tasa baja de falsos rechazos (no detectar una palabra clave cuando de hecho esta presente). Otro aspecto importante de este enfoque radica en el modelado de las partes de la señal pertenecientes a las que no son palabras clave y de hecho hay varias posibles opciones para estos MOMs de relleno.

En la elección más sencilla se modela la basura con un MOM que conecta completamente a todos los modelos de sub-unidad [Rose y Paul, 1990], mientras que la opción más compleja se modela la basura con MOMs de vocabularios grandes, en donde se excluyen del léxicon a las palabras clave [Weintraub, 1995]. Utilizando conocimientos lingüísticos adicionales este último enfoque obtiene un mejor modelo para la basura. Esta ventaja, sin embargo, induce a un mayor costo para la decodificación y requiere una gran cantidad de datos de entrenamiento, en particular para el entrenamiento del modelo de lenguaje, además de otros problemas prácticos. Desde un punto de vista conceptual uno debería preguntarse si un enfoque de detección automática debe exigir un conocimiento lingüístico tan grande. Por supuesto, existe una gama de variaciones para los modelos de la basura que se sitúan entre los dos ejemplos extremos que se han señalado anteriormente.

La decodificación Viterbi se basa en una secuencia de decisiones (a nivel local) para determinar la mejor ruta posible, lo cual puede ser frágil con respecto a posibles desajustes en el modelo local. En el contexto de la detección de palabras clave basada en MOMs, una palabra clave puede darse

por no-pronunciada si por ejemplo tan sólo su primer fonema sufre un desajuste. Para ganar robustez se han propuesto los enfoques de *proporción de verosimilitud* [Rose y Paul, 1990; Weintraub, 1995]. En esta aproximación, la puntuación o medida de confianza en la salida del detector de palabras clave es la relación o proporción entre la verosimilitud estimada por un MOM incluyendo la ocurrencia de la palabra clave objetivo y la verosimilitud estimada por un MOM excluyendo dicha ocurrencia. A continuación se realiza la detección mediante la comparación de las puntuaciones de salida con un umbral predefinido. Se han diseñado diferentes variaciones del enfoque de proporción de verosimilitud, como por ejemplo aquella en donde se calcula la proporción sólo en la parte de la señal de donde se supone que la palabra clave es detectada [Junkawitsch et al., 1997].

En general, todos los métodos descritos son variaciones del mismo paradigma basado en MOMs, el cual consiste en entrenar un modelo generativo a través de la maximización de verosimilitudes, antes de introducir diferentes modificaciones previas a la tarea de decodificación para hacer frente a la tarea de detección de palabras clave. En otras palabras, estas aproximaciones no proponen entrenar el modelo con el fin de maximizar el rendimiento de la detección y la tarea propia de la detección de palabras clave sólo se introduce hasta la etapa de inferencia, después del entrenamiento.

Sólo unos cuantos trabajos han propuesto enfoques discriminatorios de entrenamiento de parámetros para evitar esta debilidad. Véanse por ejemplo Sandness y Hetherington [2000] y Benayed et al. [2003]. Algunos de estos enfoques discriminatorios se han centrado en combinar diferentes aproximaciones para detección de palabras clave basados en MOMs. Por ejemplo entrenando una red neuronal para combinar las tasas de verosimilitud de los diferentes modelos [Weintraub et al., 1997]. O bien usando Máquinas de Vectores de Soporte para combinar los diferentes promedios de probabilidades a nivel de fono [Benayed et al., 2003].

Ambos enfoques proponen reducir al mínimo la tasa de error, que asigna pesos iguales a los dos posibles errores de detección, falsos positivos (o falsa alarma) y falsos negativos (ocurrencia de pérdida de palabras clave, a menudo llamada eliminación de palabras clave). Esta medida es, sin embargo, muy escasamente utilizada para evaluar detectores de palabras clave, debido principalmente a la naturaleza desbalanceada del problema. Aquí las palabras clave objetivo, en términos generales son muy poco frecuentes y por lo

tanto el número de posibles falsas alarmas excede por mucho al número de posibles fallas en la detección. En este caso el modelo que no se usa, el cual nunca predice la palabra clave, elimina todas las falsas alarmas y produce una tasa de error muy baja, por lo cual ningún otro sistema de detección podría competir con este modelo. Por esta razón, consideramos que el área bajo la curva ROC (o simplemente ABC) es una medida que ofrece más información y que por ello es comúnmente utilizada para evaluar eficientemente este tipo de modelos. Lograr grandes valores de ABC sería por lo tanto, un objetivo pertinente del aprendizaje para el entrenamiento discriminatorio de un detector de palabras clave.

Hasta donde conocemos, sólo en el trabajo de Chang [1995] se ha propuesto una aproximación específica para este objetivo. En dicho trabajo se presenta una metodología para maximizar el *factor de mérito* (FDM[4]), que es una analogía del ABC pero en un rango específico de tasas de falsas alarmas. Sin embargo, el enfoque propuesto se basa en diversas heurísticas, que no proveen ninguna garantía teórica para la obtención de FDM grandes. Además, estas heurísticas implican la selección de varios parámetros, que constituyen un desafío para su uso práctico.

A continuación se presenta un enfoque que tiene como objetivo lograr altos valores de ABC sobre un conjunto de ejemplos de entrenamiento y constituye una aproximación discriminatorio al problema de detección de palabras clave. El modelo propuesto se basa en técnicas de aprendizaje de margen amplio para la predicción de la secuencia y provee garantías teóricas respecto de la generalización de su desempeño. Además, el proceso de aprendizaje eficiente que involucra asegura su escalabilidad hacia problemas más grandes o de mayor complejidad y su uso práctico es simple.

6.3. Definición del problema de reconocimiento de palabras clave

Para la detección de palabras clave contamos con una señal de voz, compuesta por una secuencia de vectores de características acústicas $\overline{\mathbf{x}} = (\mathbf{x}_1, ..., \mathbf{x}_T)$ donde $\mathbf{x}_t \in \mathcal{X} \subset \mathbb{R}^d$ para todo $1 \leq t \leq T$ es un vector de características de longitud d extraído de la $t-$ésima trama. Naturalmente, la

[4]Más conocido como FOM por sus siglas del inglés Figure-Of-Merit.

longitud de la señal acústica varía de una señal a otra y por lo tanto T no es fija. Se denota a una palabra clave $k \in \mathcal{K}$, donde $\mathcal{K}$ es un lexicon de palabras. Cada palabra clave k se compone a su vez de una secuencia de fonemas $\overline{p}^k = (p_1, ..., p_L)$, en donde $p_l \in \mathcal{P}$ para todo $1 \leq l \leq L$ y $\mathcal{P}$ es el dominio de los símbolos de fonema. El número de fonemas en cada palabra clave puede variar de una palabra clave a otra y por lo tanto L no es fija. Denotamos con $\mathcal{P}^*$ (y de forma similar $\mathcal{X}^*$) al conjunto de todas las secuencias finitas en $\mathcal{P}$. Definamos también al lapso de tiempo de la secuencia de fonemas $\overline{p}^k$ en la señal de voz $\overline{\mathbf{x}}$ y denotemos como $s_l \in \{1, ..., T\}$ al tiempo de inicio (en unidades de trama) del fonema p_l en $\overline{\mathbf{x}}$.

Asumiendo que el tiempo de inicio de cualquier fonema p_l, es igual al tiempo de finalización del fonema previo p_l, es decir, $e_l = s_{l+1}$ para todo $1 \leq l \leq L-1$. Definimos al lapso de tiempo de la secuencia (o segmentación de la secuencia) como $\overline{s}_k = (s_1, ..., s_L, e_L)$. Esta notación se ilustra en la Fig. 6.2. Nuestro objetivo es entrenar a un detector de palabras clave, denotado por $f : \mathcal{X}^* \times \mathcal{P}^* \longrightarrow \mathbb{R}$, que tome como entrada el par $(\overline{\mathbf{x}}, \overline{p}^k)$ y devuelva una calificación o puntaje que exprese el nivel de confianza de que la palabra clave k dada (o palabra clave objetivo) fue pronunciada en $\overline{\mathbf{x}}$. Al comparar este resultado (o puntaje asignado) con un umbral $b \in \mathbb{R}$, podremos determinar si $\overline{p}^k$ se pronunció en $\overline{\mathbf{x}}$.

En el aprendizaje discriminatorio supervisado contamos con un conjunto de ejemplos de entrenamiento y un conjunto de prueba (o conjunto de datos para evaluación). En concreto, para la tarea de detección discriminatoria de palabras clave debemos estar provistos con dos conjuntos de palabras clave. El primer conjunto $\mathcal{K}_{entrenamiento}$ se usa para propósitos de entrenamiento y el segundo conjunto $\mathcal{K}_{prueba}$ se usa para evaluación. Es necesario tener en cuenta que el lexicon de palabras clave es la unión de ambos, el conjunto de entrenamiento y el de pruebas, entonces $\mathcal{K} = \mathcal{K}_{entrenamiento} \cup \mathcal{K}_{prueba}$.

Algorítmicamente, no restringimos la inclusión de una palabra clave en un sólo conjunto, es decir una palabra que aparece en el conjunto de entrenamiento puede aparecer también el el conjunto de pruebas. Sin embargo, para fines experimentales escogimos diferentes palabras clave para entrenamiento y prueba y por lo tanto $\mathcal{K}_{entrenamiento} \cap \mathcal{K}_{prueba} = \emptyset$.

Comúnmente los detectores de palabras clave f se evalúan el uso de la curva ROC. Esta curva es la representación gráfica de la tasa de verdaderos

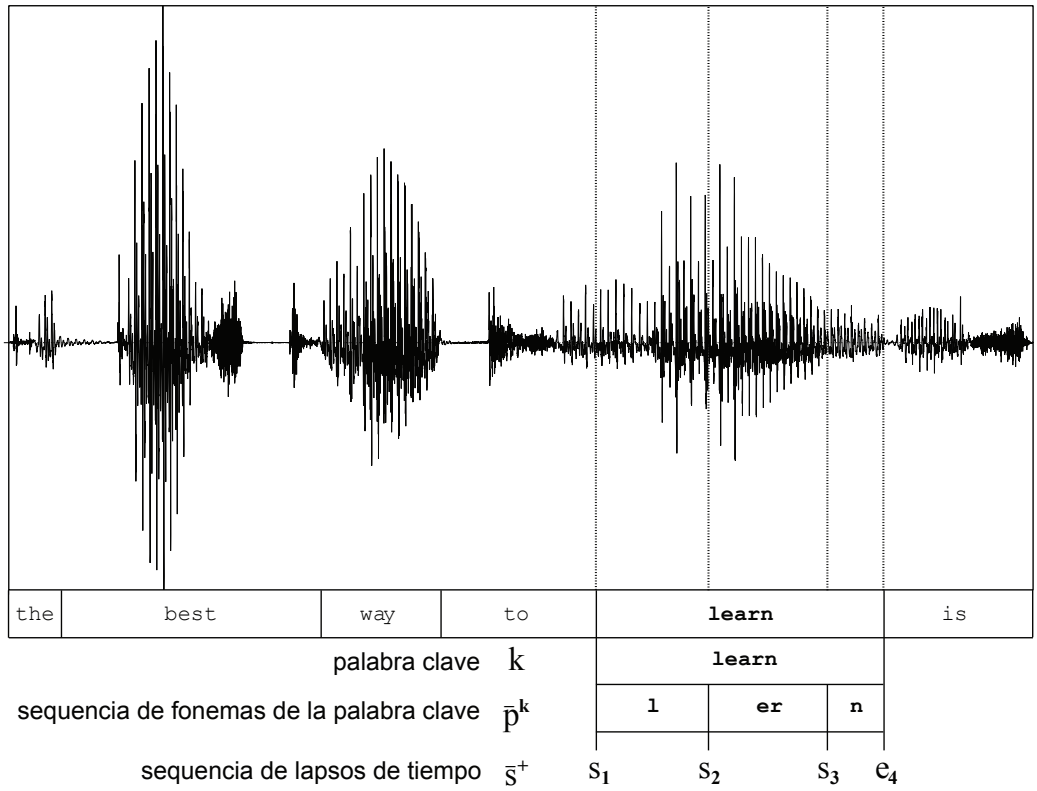

FIGURA 6.2: Ejemplo de la notación utilizada. Se muestra la forma de onda de la expresión hablada "*the best way to learn is...*" tomada del corpus TIMIT. La palabra clave k es la palabra "*learn*". La transcripción fonética de la palabra clave es $\overline{p}^k$. La secuencia de intervalos de tiempo para los fonemas de dicha transcripción es $\overline{s}^+$.

positivos (TVP) como función de la tasa de falsos positivos (TFP). La TVP mide la cantidad de ocurrencias de palabras clave detectadas correctamente, mientras que el TFP mide la cantidad de expresiones negativas (errores en la identificación de una palabra clave) dando una falsa alarma. Los puntos de la curva se obtienen recorriendo el umbral b desde la salida de mayor valor entregada por el sistema hasta el valor más pequeño. Por lo tanto, estos valores corresponden a diferentes configuraciones del equilibrio entre los dos tipos de errores que un detector de palabras clave puede cometer, es decir, fallar una expresión de palabra clave o provocar una falsa alarma. Con el fin de evaluar un detector de palabras clave con distintos ajustes de dicho

balance (entre sus ventajas y desventajas), es común reportar el ABC como un valor único. El valor de ABC por lo tanto, corresponde al rendimiento promedio del clasificador, previo a los diferentes ajustes de operación.

Una exposición sobre el análisis de sistemas de detección mediante la curva ROC o simplemente análisis ROC se realiza con más detalle en el *Apéndice B*. Sin embargo y con fines de claridad, establecemos la siguiente notación. Dada una palabra clave k, un conjunto de expresiones positivas X_k^+ en donde k es pronunciada y un conjunto de expresiones negativas X_k^- en donde k no es pronunciada, el ABC se puede escribir como

$$A_k = \frac{1}{\left|X_k^+\right|\left|X_k^-\right|} \sum_{\overline{\mathbf{x}}^+ \in X_k^+} \sum_{\overline{\mathbf{x}}^- \in X_k^-} \mathbb{1}_{\{f(\overline{p}^k, \overline{\mathbf{x}}^+) > f(\overline{p}^k, \overline{\mathbf{x}}^-)\}},$$

en donde $|\cdot|$ se refiere a la cardinalidad del conjunto y $\mathbb{1}_{\{\pi\}}$ se refiere a la *función indicadora* para la cual su valor es 1 si el predicado π se mantiene y 0 en caso contrario. El ABC de la palabra clave k, es decir A_k, estima la probabilidad de que la puntuación asignada a una expresión positiva sea mayor que la puntuación asignada a una expresión negativa[5]. Dado que también estamos interesados en conocer el rendimiento esperado para cualquier palabra clave, es común graficar la curva ROC promediada sobre un conjunto de palabras clave de evaluación $\mathcal{K}_{prueba}$, y calcular el promedio correspondiente del ABC, es decir

$$A_{prueba} = \frac{1}{|\mathcal{K}_{prueba}|} \sum_{k \in \mathcal{K}_{prueba}} A_k.$$

En este trabajo, se presenta una aproximación basada en técnicas de margen amplio para entrenar un detector de palabras clave f a partir de un conjunto de entrenamiento, con el objetivo de lograr un valor alto de ABC.

[5] A esta cantidad también se le conoce como la estadística de Wilcoxon-Mann-Whitney [Wilcoxon, 1945; Mann y Whitney, 1947; Cortes y Mohri, 2004].

6.4. Función de pérdida y parametrización del modelo

Para construir el detector palabras clave f, contamos con datos de entrenamiento que consisten en un conjunto de palabras clave de entrenamiento $\mathcal{K}_{prueba}$ y con un conjunto de expresiones de entrenamiento. Para cada palabra clave $k \in \mathcal{K}_{prueba}$, denotamos con X_k^+ al conjunto de expresiones en las cuales la palabra clave se pronuncia y con X_k^- al conjunto del resto de las expresiones, en las cuales la palabra clave no se pronuncia. Además para cada expresión positiva $\overline{\mathbf{x}}^+ \in X_k^+$, contamos con la secuencia de tiempo $\overline{s}^+$ para la secuencia de fonemas de la palabra clave $\overline{p}^k$ en $\overline{\mathbf{x}}^+$. Tal secuencia de fonemas nos indica los puntos de inicio y fin para cada uno de los fonemas de la palabra clave y puede incluso ser suministrado por anotaciones manuales o bien haciendo uso de algún algoritmo o técnica automatizada de alineamiento [6]. Si definimos al conjunto de datos de entrenamiento como,

$$\mathcal{T}_{entrenamiento} \equiv \{(p^{k_i}, \overline{\mathbf{x}}_i^+, \overline{s}_i^+, \overline{\mathbf{x}}_i^-)\}_{i=1}^m.$$

Para cada palabra clave en el conjunto de datos de entrenamiento existe sólo una expresión positiva y una expresión negativa, por lo tanto $\left|X_k^+\right| = 1$, $\left|X_k^-\right| = 1$ y además $|\mathcal{K}_{entrenamiento}| = m$. Entonces el ABC para el conjunto de entrenamiento se convierte en,

$$A_{entrenamiento} = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{\{f(\overline{p}^{k_i}, \overline{\mathbf{x}}_i^+) > f(\overline{p}^{k_i}, \overline{\mathbf{x}}_i^-)\}}.$$

La selección de un modelo de maximización de esta ABC es equivalente a minimizar la pérdida expresada como,

$$\mathcal{L}^{0/1}(f) = 1 - A_{entrenamiento} = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{\{f(\overline{p}^{k_i}, \overline{\mathbf{x}}_i^+) > f(\overline{p}^{k_i}, \overline{\mathbf{x}}_i^-)\}}.$$

Sin embargo la expresión de la perdida $\mathcal{L}^{0/1}$ no es adecuada para el entrenamiento del modelo, ya que es una cantidad combinatoria que es difícil de minimizar directamente. En lugar de ello adoptamos una estrategia común

[6] Como el algoritmo descrito en el *Capítulo 4* sobre segmentación de fonemas.

en los clasificadores de margen amplio y empleamos la *función de pérdida en forma de bisagra convexa*[7], es decir

$$\mathcal{L}(f) = \frac{1}{m}\sum_{i=1}^{m} \left[1 - f(\overline{p}^{k_i}, \overline{\mathbf{x}}_i^{+}) + f(\overline{p}^{k_i}, \overline{\mathbf{x}}_i^{-})\right]_+, \tag{6.1}$$

en donde $[a]_+$ denota a máx$\{0, a\}$. Los límites superiores $\mathcal{L}^{0/1}(f)$ de la función de pérdida en forma de bisagra $\mathcal{L}(f)$, dado que para cualquier número a y b,

$$[1 - a + b]_+ \geq 1_{\{a \leq b\}},$$

y por otra parte, cuando $\mathcal{L}(f) = 0$, entonces $A_{entrenamiento} = 1$. Además sabemos que para cualquier a y b,

$$[1 - a + b]_+ = 0 \implies a > b + 1 \implies a > b.$$

La función de pérdida en forma de bisagra está relacionada con la pérdida durante la clasificación usada tanto en las MVS para regresión ordinal como en la clasificación con MVS. Estos enfoques han demostrado ser exitosos en problemas altamente desbalanceados, como por ejemplo en la recuperación de la información [Grangier y Bengio, 2008], por lo tanto el uso de la función de pérdida en forma de bisagra para el problema de detección de palabras clave resulta útil y atractiva. Es posible parametrizar al detector de palabras clave f como,

$$f_{\mathbf{w}}(\overline{\mathbf{x}}, \overline{p}^k) = \max_{\overline{s}} \mathbf{w} \cdot \phi(\overline{\mathbf{x}}, \overline{p}^k, \overline{s}), \tag{6.2}$$

en donde $\mathbf{w} \in \mathbb{R}^n$ es un vector de pesos de importancia y $\phi(\overline{\mathbf{x}}, \overline{p}^k, \overline{s})$ es un vector de la función característica que mide las diferentes características relacionadas con el valor de confianza de que la secuencia de fonemas $\overline{p}^k$, que representa a la palabra clave k, sea pronunciada en $\overline{\mathbf{x}}$ en el intervalo de tiempo $\mathbf{s}$. Formalmente ϕ es una función definida como $\phi : \mathcal{X}^* \times (\mathcal{P}^* \times \mathbb{N})^* \longrightarrow \mathbb{R}^n$ Durante esta parte del trabajo se utilizaron siete *funciones características* ($n = 7$), que son similares a las empleadas en el *Capítulo 4* sobre la segmentación de fonemas y que se describen brevemente a continuación:

A cuatro de ellas las llamamos *funciones de transición de fonemas*, y tienen como objetivo la detección de transiciónes entre cada fonema. Para

[7]Mejor conocida como *convex hinge loss function*.

ello, calculan la distancia entre tramas antes y después de hacer una hipótesis respecto del punto de transición entre ellos. Formalmente, decimos que,

$$\phi(\overline{\mathbf{x}}, \overline{p}^k, \overline{s}) = \frac{1}{L}\sum_{j=2}^{L-1} d(x_{s_{j-i}}, x_{s_{j+1}}), \quad \forall i = 1, 2, 3, 4, \tag{6.3}$$

en donde d se refiere a la distancia Euclidiana y L se refiere a el número de fonemas en la palabra clave k.

La siguiente es la *función de clasificación de fonemas basada en tramas*, y se basa en un clasificador para medir la correspondencia entre cada trama y la clase de fonema sobre la cual se hizo una hipótesis,

$$\phi_5(\overline{\mathbf{x}}, \overline{p}^k, \overline{s}) = \frac{1}{L}\sum_{i=1}^{L}\sum_{t=s_i}^{s_{i+1}-1} \frac{1}{s_{i+1} - s_i} g(\mathbf{x}_t, p_i) \tag{6.4}$$

en donde $g : \mathcal{X} \times \mathcal{P} \to \mathbb{R}$ se refiere al clasificador de fonemas, el cual regresa un valor de confianza de que el vector de características acústicas en la t-ésima trama $\mathbf{x}_t$, represente a un fonema específico p_i.

La sexta función a la que llamamos *función de duración del fonema* mide la calidad de la hipótesis acerca de la segmentación del fonema $\overline{s}$, con respecto al modelo de duración, y se representa como

$$\phi_6(\overline{\mathbf{x}}, \overline{p}^k, \overline{s}) = \frac{1}{L}\sum_{i=1}^{L} \log \mathcal{N}(s_{i+1} - s_i; \mu_{p_i}, \sigma^2_{p_i}), \tag{6.5}$$

en donde $\mathcal{N}$ denota la verosimilitud del modelo de duración Gaussiano, cuyos parámetros de media μ_p y varianza σ^2_p para cada fonema p se estiman para los datos de entrenamiento.

La séptima función a la que llamamos *función de velocidad de habla* cuantifica la estabilidad para un determinado tipo de habla,

$$\phi_7(\overline{\mathbf{x}}, \overline{p}^k, \overline{s}) = \frac{1}{L}\sum_{i=2}^{L} (r_i - r_{i-1})^2, \tag{6.6}$$

en donde r_i denota la estimación acerca de la velocidad del habla para el $i-$ésimo fonema, es decir

$$r_i = \frac{s_{i+1} - s_i}{\mu_{p_i}}.$$

Este conjunto de siete funciones se utiliza para fines experimentales. Por supuesto, dicho conjunto puede ampliarse para incorporar características adicionales, tales como valores de confianza para un clasificador basado en tramas con trifonemas o para la salida de un reconocedor basado en un mejor modelo de duración.

En resumen, el detector de palabras clave presentado, genera un puntaje (sobre la medida de confianza) mediante la maximización de la suma ponderada de las funciones de características a lo largo de todos los lapsos de tiempo posibles. Esta maximización corresponde a la búsqueda en un número exponencialmente grande de lapsos de tiempo. Sin embargo este cálculo, puede realizarse de manera eficiente mediante la selección de *funciones características separables*, que permitan la aplicación de técnicas de programación dinámica, tal como los utilizados por los enfoques basados en MOMs [Rabiner y Juang, 1993].

6.5. Algoritmo iterativo de entrenamiento

En esta parte del trabajo se describe un algoritmo iterativo para encontrar el vector de pesos $\mathbf{w}$. Se puede mostrar que el vector de pesos $\mathbf{w}$ que se encuentra durante este proceso minimiza la pérdida $\mathcal{L}(f_{\mathbf{w}})$. Es decir minimiza $\mathcal{L}^{0/1}$ y a su vez da lugar a un detector de palabras clave que alcanza valor grande para el área bajo la curva (ABC) para el conjunto de datos de entrenamiento. También se muestra que el vector de pesos aprendido tiene buenas propiedades de generalización en el conjunto de datos de de pruebas.

El procedimiento comienza iniciando al vector de pesos en cero (con el vector nulo), $\mathbf{w}_0 = 0$. Luego en la iteración $i \geq 1$ el algoritmo examina el $i-$ésimo ejemplo de entrenamiento $(\overline{p}^{k_i}, \overline{\mathbf{x}}_i^+, \overline{s}_i^+, \overline{\mathbf{x}}_i^-)$. Primero el algoritmo predice el mejor lapso de tiempo para la secuencia de fonemas de la palabra clave $\overline{p}^{k_i}$ en la expresión negativa $\overline{\mathbf{x}}_i^-$ (la expresión que no se pronuncia),

$$\overline{s}_i^- = \operatorname*{argmax}_{\overline{s}} \mathbf{w}_{i-1} \cdot \phi(\overline{\mathbf{x}}_i^-, \overline{p}_i^k, \overline{s}). \tag{6.7}$$

Entonces, tomando en cuenta la pérdida en el $i-$ésimo ejemplo de entrenamiento el algoritmo comprueba que la diferencia entre la puntuación asignada a la expresión positiva y la puntuación asignada al ejemplo negativo sea mayor que 1. Formalmente, decimos que

$$\Delta\phi_i = \phi(\overline{\mathbf{x}}_i^+, \overline{p}_i^k, \overline{s}_i^+) - \phi(\overline{\mathbf{x}}_i^-, \overline{p}_i^k, \overline{s}_i^-). \tag{6.8}$$

si $\mathbf{w}_{i-1} \cdot \Delta\phi_i \geq 1$, el algoritmo mantiene el vector de pesos para la siguiente iteración, es decir, $\mathbf{w}_i = \mathbf{w}_{i-1}$. De lo contrario, el algoritmo actualiza el vector de pesos para minimizar la siguiente expresión que representa un problema de optimización

$$\mathbf{w}_i = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{w} - \mathbf{w}_{i-1}\|^2 + c\,[1 - \mathbf{w} \cdot \Delta\phi_i]_+ , \tag{6.9}$$

en donde el parámetro $c \geq 1$ controla el ajuste del balance entre mantener al nuevo vector de pesos cercano al vector anterior y satisfacer la restricción para el ejemplo actual. La Ec. 6.9 puede resolverse analíticamente si tenemos en cuenta que $\mathbf{w}_i = \mathbf{w}_{i-1} + \alpha_i \Delta\phi_i$, en donde

$$\alpha_i = \text{mín} \left\{ c, \frac{[1 - \mathbf{w}_{i-1} \cdot \Delta\phi_i]_+}{\|\Delta\phi_i\|^2} \right\}. \tag{6.10}$$

Esta modificación es considerada una estrategia *pasiva-agresiva*, ya que el algoritmo "pasivamente" mantiene el peso anterior ($\mathbf{w}_i = \mathbf{w}_{i-1}$) si la pérdida para el ejemplo de entrenamiento actual ya es cero en ese instante $\big([1 - \mathbf{w}_{i-1} \cdot \Delta\phi_i]_+ = 0\big)$, en tanto que de otra manera el algoritmo "agresivamente" actualiza el vector de pesos para compensar esta pérdida. Al final del procedimiento de entrenamiento, cuando todos los ejemplos (de entrenamiento) han sido visitados, el mejor vector de pesos $\mathbf{w}$ de entre $\{\mathbf{w}_0, ..., \mathbf{w}_m\}$ se selecciona de un conjunto de ejemplos para validación $T_{validacin}$. El parámetro c también debe seleccionarse de tal manera que optimice el desempeño en los datos de validación. El pseudocódigo del *Algoritmo 3* se muestra más adelante.

6.6. Resultados experimentales

Para evaluar la efectividad del método discriminatorio presentado llevamos a cabo experimentos usando el corpus TIMIT [Garofolo et al., 1993], el cual está compuesto de expresiones habladas en el idioma inglés, leídas por 630 hablantes provenientes de Estados Unidos de Norte América, donde cada hablante leyó 10 expresiones distintas[8]. El corpus está provisto del alineamiento y etiquetado manual a nivel de palabra y de fonema y adicionalmente de transcripciones para cada expresión hablada.

El corpus también proporciona una división estándar en dos conjuntos de datos, uno para entrenamiento y otro para pruebas. Para todos los experimentos, tanto el sistema discriminatorio como en el sistema basado en MOMs, fueron entrenados con datos de dicho corpus. Dividimos la porción de entrenamiento del corpus (excluyendo las frases contenidas en SA1 y SA2) en dos subconjuntos de datos disjuntos que contienen 500 y 3196 expresiones habladas.

La primera parte del conjunto de datos de entrenamiento, con 500 frases, se usó para entrenar las funciones g_p que se describen en la Ec. 6.4 y que definen a la función característica ϕ_5 o *función de clasificación de fonemas basada en tramas*. Estas funciones fueron aprendidas usando el algoritmo descrito en el *Capítulo 3* sobre la clasificación de fonemas, usando los MFCC[9], junto con sus primeras y segundas derivadas MFCC$+\Delta+\Delta\Delta$ como vectores de características acústicas[10] y un núcleo de tipo Gaussiano con $\sigma = 6{,}0$ como parámetro. Usando las funciones g_p como clasificador de fonemas obtuvimos una precisión del **53 %** por trama, usando el conjunto de datos de prueba del corpus TIMIT.

La segunda parte del conjunto con 3196 frases, conformó el conjunto de datos de entrenamiento para el detector de palabras clave. De este conjunto elegimos 200 palabras clave al azar para entrenamiento y 200 palabras clave distintas para validación, a partir del diccionario del corpus TIMIT. Las

[8]Una descripción más detallada del corpus del habla continua TIMIT se puede ver en el *Apéndice A* del presente trabajo.

[9]Coeficientes cepstrales en las frecuencias Mel.

[10]Obtenidos en forma estándar basándonos en el estándar ETSI para reconocimiento de voz distribuido [ES-202-211].

Algoritmo 3 Algoritmo iterativo de entrenamiento para detección de palabras clave

Entrada: Conjunto de datos de entrenamiento $T_{entrenamiento}$.

Conjunto de datos de validación.

Parámetro c.

Salida: $\mathbf{w}$ que logra el más alto valor de ABC en el conjunto $\mathcal{T}_{validación}$:

$$\mathbf{w} = \underset{\mathbf{w}\in\{\mathbf{w_1},\ldots,\mathbf{w_m}\}}{\operatorname{argmin}} \frac{1}{m_{validación}} \sum_{j=1}^{m_{validación}} \mathbb{1}_{\{\text{máx}_{\bar{s}^+}\ \mathbf{w}\cdot\phi(\overline{\mathbf{x}}_j^+,\overline{p}^{k_j},\overline{s}^+)>\text{máx}_{\bar{s}^+}\ \mathbf{w}\cdot\phi(\overline{\mathbf{x}}_j^-,\overline{p}^{k_j},\overline{s}^-)\}}$$

Inicializar: $\mathbf{w}_0 = 0$.

Repetir

Para cada $(p^{k_i}, \overline{\mathbf{x}}_i^+, \overline{s}_i^+, \overline{\mathbf{x}}_i^-) \in \mathcal{T}_{entrenamiento}$:

Hacer $\overline{s}_i^- = \arg\text{máx}_{\bar{s}}\ \mathbf{w}_{i-1}\cdot\phi(\overline{\mathbf{x}}_i^-, \overline{p}^{k_i}, \overline{s})$

Hacer $\Delta\phi_i = \phi(\overline{\mathbf{x}}_i^+, \overline{p}^{k_i}, \overline{s}_i^+) - \phi(\overline{\mathbf{x}}_i^-, \overline{p}^{k_i}, \overline{s}_i^-)$.

Si $\mathbf{w}_{i-1}\cdot\Delta\phi_i < 1$ **entonces**

$$\alpha_i = \text{mín}\left\{c, \frac{1-\mathbf{w}_{i-1}\cdot\Delta\phi_i}{\|\Delta\phi_i\|^2}\right\}$$

Actualizar: $\mathbf{w}_i = \mathbf{w}_{i-1} + \alpha_i\cdot\Delta\phi_i$

fin Si

fin Repetir

palabras clave que se eligieron tienen una longitud mínima de 6 fonemas. Para cada una de las palabras clave elegimos una *expresión positiva* en la que se pronunció la palabra clave, y una *expresión negativa* en la que la palabra clave no se pronunció. La misma frase podría ser una expresión positiva para una palabra clave y una expresión negativa de una palabra clave diferente, en cualquier caso las expresiones habladas utilizadas para el entrenamiento no se utilizaron para la validación.

Ejecutamos el algoritmo iterativo discriminatorio con un valor de parámetro $c = 1$. El algoritmo discriminatorio probó ser robusto para el conjunto de entrenamiento y para el conjunto de palabras clave de validación. Además como resultado de la elección de conjuntos diferentes se obtuvieron resultados con valores de rendimiento similares.

Dado que el método presentado es independiente del contexto y con la finalidad de realizar una comparación con un sistema basado en MOMs, entrenamos un reconocedor de fonemas también independiente del contexto, basado en modelos ocultos de Markov. Para ello usamos el conjunto completo de entrenamiento del corpus TIMIT. Usamos 3600 frases como conjunto de entrenamiento y 100 frases como conjunto de validación. Usamos los mismos coeficientes MFCC+$\Delta + \Delta\Delta$ como vectores de características acústicas, normalizados usando el *método de normalización en media y varianza*[11].

En la configuración seleccionada cada fonema se representó con un MOM de 5 estados emitidos, en una topología de izquierda a derecha y con 40 mezclas Gaussianas (véase Young et al. [2006]). Todos los experimentos basados en MOMs se realizaron usando HTK[12] (Hidden Markov Model Toolkit) que es un grupo de herramientas de software para construcción y manipulación de MOMs. Usando el conjunto de datos de prueba del corpus TIMIT, obtuvimos como resultado un MOM independiente del contexto para reconocimiento de fonemas que alcanzó una precisión del 60,3 %.

[11]Mejor conocido como CMVN, por sus siglas del inglés *cepstral mean and variance normalization.*

[12]Se utiliza principalmente para la investigación del reconocimiento automático del habla, aunque se ha utilizado en otras numerosas aplicaciones, incluyendo la investigación de síntesis de voz. Sus herramientas proporcionan sofisticadas capacidades para el análisis del habla, entrenamiento de MOMs, pruebas y análisis de resultados. El software es compatible tanto con los MOMs de mezclas de funciones Gaussianas de densidad continua, como distribuciones discretas y se puede utilizar para construir sistemas complejos [HTK].

En el caso del sistema basado en MOMs, la detección de palabras clave se realizó mediante el uso de un modelo de Markov adicional, compuesto a su vez por dos MOMs, un modelo de palabras clave y un modelo de basura, como se ilustra en la Fig. 6.1. El modelo de la palabra clave es un MOM que calcula la probabilidad de una secuencia acústica, dado que ésta representa la secuencia de fonemas de la palabra clave. El modelo para basura se compone de los modelos de fonemas totalmente conectados entre sí, que estiman la probabilidad de cualquier secuencia acústica. El MOM global conecta completamente al modelo de palabras clave y al modelo de basura.

La detección de una determinada palabra clave, en una expresión hablada proveniente del conjunto de datos de prueba, se realizó a través de una búsqueda de la mejor ruta, en donde se introdujo un parámetro externo de la probabilidad *a priori* de la palabra clave al sub-modelo oculto de Markov de la palabra clave. El mejor camino encontrado por el algoritmo Viterbi en el MOM global, o bien pasa a través del modelo de la palabra clave (en cuyo caso se dice que la palabra clave fue pronunciada) o no (en cuyo caso la palabra clave no se pronunció en esa secuencia acústica). Mediante la configuración de los parámetros de la probabilidad *a priori* se ajustó el equilibrio entre *la tasa de verdaderos positivos* y *la tasa de falsos positivos*.

El conjunto de pruebas se construyó con 100 palabras clave seleccionadas aleatoriamente y que son distintas a las palabras clave elegidas para los conjuntos de datos de validación y entrenamiento. A partir del diccionario del corpus, se seleccionaron palabras clave con una longitud mínima de cuatro fonemas. Para cada palabra clave, seleccionamos al azar un máximo de 20 frases en las que se pronunció la palabra y un máximo de 20 expresiones en las que no se pronunció. Hay que tomar en cuenta que el número de expresiones habladas en las que se pronunció la palabra clave no es siempre igual a 20, ya que algunas palabras se pronuncian menos de 20 veces en el conjunto completo de datos de prueba del corpus TIMIT. Tanto el algoritmo discriminatorio como el algoritmo basado en MOMs fueron evaluados en este conjunto de datos de prueba.

Reportamos los resultados de los experimentos realizados, mediante el uso de las curvas ROC que se muestran en la Fig. 6.3. Las áreas bajo las curvas ROC tienen un valor de **0.9925** y **0.93616** para el algoritmo discriminatorio y el algoritmo basado en MOMs respectivamente. Los resultados

obtenidos demostraron que el modelo discriminatorio presentado ofrece consistentemente un mejor desempeño en la tarea de detección del conjunto de palabras clave seleccionadas.

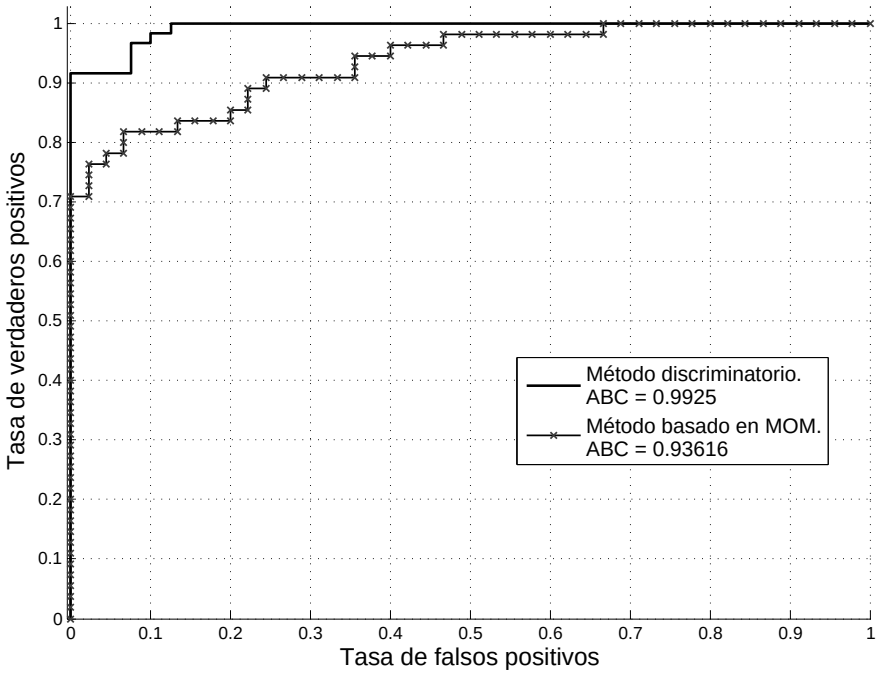

Figura 6.3: Curvas ROC para el algoritmo discriminatorio y el enfoque basado en MOM, entrenados con el conjunto de datos de entrenamiento del corpus TIMIT y probados con 100 palabras clave provenientes del conjunto de datos de prueba del mismo corpus. Las áreas bajo las curvas ROC tienen un valor de **0.9925** y **0.93616** para los algoritmos discriminatorio y basado en MOM respectivamente.

Para asegurarnos de que el procedimiento presentado puede aplicarse en ausencia de conjuntos de datos alineados manualmente, se entrenó un modelo usando los intervalos de tiempo obtenidos del conjunto de datos alineados usando el algoritmo automático de segmentación presentado en el *Capítulo 4* sobre *segmentación de fonemas*. Dicho algoritmo automático fue entrenado con 50 frases del conjunto de datos de entrenamiento del corpus, que no fueron utilizadas para el entrenamiento o la validación del algoritmo de detección

	Discriminatorio	**Basado en MOM**
Área bajo la curva ROC	**0.9925**	0.93616

Tabla 6.1: Comparación de valores de área bajo la curva ROC obtenidos por los métodos discriminatorio y basado en MOM.

de palabras clave. El ABC obtenida por algoritmo discriminatorio entrenado con datos provenientes del alineamiento automático fue aproximadamente **0.9924**, es decir prácticamente idéntica al ABC del algoritmo entrenado con datos del alineamiento obtenido manualmente.

Un resumen de los resultados obtenidos en todos los experimentos se presentan en las Tablas 6.1 y 6.2. Un análisis más detallado de estos muestra que el algoritmo discriminatorio supera al enfoque basado en MOMs sistemáticamente en términos del ABC obtenida. Este hecho confirma la hipótesis de que maximizar el área bajo la curva ROC es una buena estrategia para mejorar el desempeño de un sistema de detección de palabras clave. Por otra parte, el algoritmo discriminatorio supera al algoritmo basado en MOMs en casi todos los puntos de la curva ROC, lo que significa que tiene una mejor tasa de verdaderos positivos para todas las tasas de falsos negativos dadas.

Alineamiento	**Discriminatorio**	**Basado en MOM**
Manual	**0.9925**	0.93616
Automático	**0.9924**	0.936

Tabla 6.2: Comparación de resultados del entrenamiento del modelo discriminatorio usando el alineamiento manual de fonemas del corpus TIMIT y el alineamiento automático. El ABC del modelo discriminatorio usando el conjunto de datos de prueba del corpus TIMIT se compara con los resultados obtenidos con el modelo basado en MOM.

Capítulo 7
Conclusiones y trabajo futuro

En este trabajo se presentan algoritmos que establecen las bases para un modelo acústico basado en métodos de margen amplio para el reconocimiento automático del habla continua. Por medio de los experimentos realizados usando el corpus TIMIT, hemos mostrado la utilidad de los diferentes aspectos de tal modelo acústico. A lo largo del trabajo de investigación, se presentan los fundamentos teóricos que sustentan los algoritmos propuestos y se demuestra que reducen al mínimo la pérdida asociada con cada tarea, tanto en el conjunto de datos de entrenamiento como en los conjuntos de datos de prueba.

Comenzando con la clasificación jerárquica de fonemas. Se presentó un clasificador multiclase basado en tramas, que impone una noción de "gravedad" a los errores de predicción, de acuerdo con una estructura jerárquica de fonemas predefinida. Se describen dos algoritmos, uno en línea y otro de procesamiento por lotes para resolver el problema de optimización con restricciones que representa la clasificación multiclase de fonemas organizados en una estructura jerárquica. Se analiza la pérdida asociada al peor de los casos para el algoritmo en línea y una posible generalización de dicho algoritmo a su versión en modo de procesamiento por lotes.Se muestran las

fortalezas del enfoque presentado mediante una serie de experimentos usando un corpus fonético-acústico del habla continua estándar, así como mediante la aplicación del algoritmo a un conjunto de datos sintéticos. Los resultados indicaron claramente la validez de la hipótesis formulada respecto del uso de la estructura jerárquica de fonemas, es decir, que su uso y explotación resul-

ta ventajosa para disminuir errores en la tarea de clasificación de fonemas. En todos los experimentos, tanto en la versión del algoritmo en línea como por lotes, el clasificador jerárquico de fonemas logró una disminución en los valores del error asociado a la tarea con respecto a su contraparte "plana", es decir aquella que es ajena a la estructura jerárquica propuesta. A su vez el algoritmo de clasificación jerárquica de fonemas presentado, sirvió como base para dos algoritmos adicionales uno para alineamiento de fonemas y otro para el reconocimiento de palabras clave.

A continuación se presentó un algoritmo discriminatorio para aprendizaje de una función de alineamiento de una señal de voz con su respectivo fonema a partir de un conjunto de datos de ejemplo para entrenamiento. La contribución de dicho algoritmo es en dos sentidos. Por una parte muestra cómo puede ser utilizado un algoritmo basado en técnicas de margen amplio para modelar el problema del alineamiento de una señal de voz con su correspondiente fonema. En segundo lugar, ejemplifica la construcción de un algoritmo simple y efectivo para resolver dicho problema. Este algoritmo de aprendizaje es más eficiente que los algoritmos similares (documentados en la literatura respecto al alineamiento de fonemas) para la predicción de secuencias con aproximaciones de margen amplio y su aplicación resulta es más adecuada para el problema del reconocimiento del habla, para el que comúnmente se dispone de un gran número de ejemplos de entrenamiento. Adicionalmente el algoritmo de aprendizaje es fácil de implementar y ofrece garantías de convergencia.

Por otra parte, se muestra evidencia teórica y empírica que demuestra la capacidad de generalización de este método. De hecho, los resultados obtenidos de los experimentos sugieren que el entrenamiento discriminatorio requiere menos ejemplos de entrenamiento que los procedimientos de alineamiento automático basados en modelos ocultos de Markov. De los experimentos realizados con el corpus TIMIT concluimos que el método presentado para alineamiento de fonemas compite con otras aproximaciones consideradas actualmente parte del estado del arte.

Se presenta un algoritmo de reconocimiento de secuencias completas de fonemas que está basado en aprendizaje supervisado discriminatorio y núcleos de Mercer. Dado que el error en el reconocimiento de fonemas comúnmente se mide en términos de la distancia de Levenshtein, es decir la mínima distancia de edición (de operaciones necesarias para transformar la secuencia de

fonemas predicha en la verdadera) se presentó un algoritmo que minimiza directamente dicha distancia para un conjunto de datos de entrenamiento con frases de ejemplo transcritas y se mostró el rendimiento de su generalización en un conjunto de datos distinto (oculto).

Es importante aclarar que hasta ahora, los resultados experimentales obtenidos con el método discriminatorio, para la tarea del reconocimiento de fonemas, son inferiores a los resultados obtenidos por las aproximaciones basadas en modelos ocultos de Markov más recientes. Sin embargo, mientras que por una parte ha habido un enorme esfuerzo continuo en el uso de modelos ocultos de Markov para el reconocimiento de secuencias de fonemas, el método discriminatorio es por otro lado bastante innovador y la elección de las características así como de los operadores del núcleo para el reconocimiento del habla no ha sido estudiado de forma integral. Es por ello que consideramos necesario, como trabajo a futuro, la experimentación con nuevas características y operadores, antes de poder utilizar y aprovechar todas las ventajas de los métodos basados en núcleo.

Por último, se presentó un algoritmo para el reconocimiento de palabras clave basado en un enfoque discriminatorio. Si bien es cierto que hoy en día el estado del arte en la técnica de reconocimiento automático del habla se basa sobre todo en los enfoques generativos tradicionales, basados en modelos ocultos de Markov, se introdujo un enfoque discriminatorio que en lugar de modelar la distribución de los datos de entrada asumiendo una clase destino, se ocupa de modelar la distribución de la clase posterior, maximizando la discriminación entre objetivos similares acústicamente.

Este enfoque discriminatorio para el reconocimiento de palabras clave, directamente optimiza la función relacionada con el área bajo la curva ROC, es decir, la medida más común para la evaluación de detectores de palabras clave. En comparación con los enfoques generativos basados en modelos ocultos de Markov, el modelo presentado mostró una mejora estadísticamente significativa en el rendimiento, usando el corpus TIMIT. Los algoritmos presentados en este trabajo establecen elementos básicos para el reconocimiento del habla continua basada en métodos de margen amplio.

Sin embargo, en la aplicación de los métodos de margen amplio al reconocimiento del habla, sigue habiendo un déficit que debe remediarse ya que

estos algoritmos no son lo suficientemente robustos para el caso del reconocimiento del habla continua con gran vocabulario. Es decir para construir un sistema que pueda reconocer lenguaje espontáneo con vocabulario de un tamaño igual o mayor a unas $10,000$ palabras. Para ello es necesario extender las nociones presentadas en este trabajo a tal reconocimiento. También es necesario encontrar un método para incluir información contextual (como modelos de lenguaje) en el modelo discriminatorio presentado. Adicionalmente y para tal propósito sera necesario trabajar en la reducción del costo computacional que conllevan los algoritmos basados en núcleo.

Apéndice A
Corpus acústico-fonético del habla continua TIMIT

El corpus acústico-fonético del habla continua TIMIT (Texas Instruments (**TI**) y Massachusetts Institute of Technology (**MIT**)) [Garofolo et al., 1993] descrito en [Zue et al., 1990], contiene grabaciones de expresiones habladas, fonéticamente balanceadas y pronunciadas en idioma inglés. Fueron grabadas usando un micrófono Sennheiser (para voz cercana), a una frecuencia de 16 kHz y con una resolución de 16 bits por muestra.

TIMIT contiene un total de 6300 frases (5,4 horas), y está compuesta de 10 frases pronunciadas por cada uno de los 630 hablantes provenientes de las 8 principales regiones dialectales de los Estados Unidos de América. Todas las frases fueron etiquetadas manualmente a nivel de palabra y fonema. De las 6300 expresiones habladas, 2 son frases dialectales (SA), 450 son frases fonéticamente compactas (SX) y 1890 son frases fonéticamente diversas (SI).

La documentación del corpus TIMIT propone conjuntos de datos para entrenamiento ($\approx 70\,\%$) y prueba, de la forma en que se describe en la Tabla A.1. El conjunto de datos de prueba contiene 4620 expresiones habladas, pero usualmente sólo se utilizan las frases SI y SX, por lo cual dicho conjunto consta de 3696 frases pronunciadas por 462 hablantes. El conjunto de datos de prueba contiene 1344 expresiones pronunciadas por 168 hablantes.

El conjunto núcleo o básico de pruebas, que es la versión abreviada del conjunto completo de pruebas, consiste en 192 expresiones habladas, 8 por

Conjunto de datos	# **hablantes**	# **frases**	# **horas**
de entrenamiento	462	3696	3.14
básico de pruebas	24	192	0.16
completo de pruebas	168	1344	0.81

Tabla A.1: Descripción de los conjuntos de datos de entrenamiento y pruebas del corpus TIMIT.

cada 24 hablantes (2 hombres y 1 mujer de cada región dielectal). Con excepción de las frases en el subconjunto SA, que usualmente se excluyen del conjunto de datos de prueba, los conjuntos de entrenamiento y de prueba no se superponen.

Las transcripciones originales del corpus TIMIT se basan en 61 fonemas, que se presentan en la Tabla B.1. El alfabeto usado se conoce como TIMIT-BET y éste está inspirado en el alfabeto ARPABET. Los detalles sobre la transcripción y el alineamiento manual de los fonemas puede encontrarse en [Zue y Seneff, 1996] y los detalles sobre el análisis fonético pertinente puede consultarse en [Keating et al., 1994]. Los 61 fonemas del corpus TIMIT algunas veces se consideran una descripción demasiado limitada para su uso práctico. Sin embargo otros autores, para fines de entrenamiento han compactado al conjunto de 61 fonemas en sólo 48. Para efectos de evaluación, las 61 etiquetas del corpus TIMIT comúnmente se compactan en un conjunto de 39 fonemas, según lo propuesto por Lee y Hon [Lee y Hon, 1989].

El corpus del habla TIMIT ha sido la base de datos estándar para la comunidad de reconocimiento del habla durante varias décadas y sigue siendo ampliamente utilizado hoy en día, tanto para el habla como para experimentos de reconocimiento del locutor. Esto no sólo se debe a que cada enunciado está etiquetado manualmente a nivel fonético y están provistos del códigos correspondiente al número del hablante, sexo y región dialectal, sino también porque se considera lo suficientemente pequeño como para garantizar un tiempo de respuesta relativamente rápida para los experimentos completos y lo suficientemente grande como para demostrar las capacidades, características y desempeño de los sistemas.

	Fono	Ejemplo		Fono	Ejemplo		Fono	Ejemplo
1	iy	b*ee*t	**22**	ch	*ch*oke	**43**	en	butto*n*
2	ih	b*i*t	**23**	b	*b*ee	**44**	eng	Wash*ing*ton
3	eh	b*e*t	**24**	d	*d*ay	**45**	l	*l*ay
4	ey	b*ai*t	**25**	g	*g*ay	**46**	r	*r*ay
5	ae	b*a*t	**26**	p	*p*ea	**47**	w	*w*ay
6	aa	b*o*b	**27**	t	*t*ea	**48**	y	*y*acht
7	aw	b*ou*t	**28**	k	*k*ey	**49**	hh	*h*ay
8	ay	b*i*te	**29**	dx	mu*dd*y	**50**	hv	a*h*ead
9	ah	b*u*t	**30**	s	*s*ea	**51**	el	bott*le*
10	ao	b*ough*t	**31**	sh	*sh*e	**52**	bcl	b closure
11	oy	b*oy*	**32**	z	*z*one	**53**	dcl	d closure
12	ow	b*oa*t	**33**	zh	a*z*ure	**54**	gcl	g closure
13	uh	b*oo*k	**34**	f	*f*in	**55**	pcl	p closure
14	uw	b*oo*t	**35**	th	*th*in	**56**	tcl	t closure
15	ux	t*oo*t	**36**	v	*v*an	**57**	kcl	k closure
16	er	b*ir*d	**37**	dh	t*h*en	**58**	q	glotal stop
17	ax	*a*bout	**38**	m	*m*o*m*	**59**	pau	pause
18	ix	deb*i*t	**39**	n	*n*oo*n*	**60**	epi	epenthetic
19	axr	butt*er*	**40**	ng	si*ng*	**61**	h#	begin/end
20	ax-h	*su*spect	**41**	em	bott*om*			
21	jh	*j*oke	**42**	nx	wi*nn*er			

Tabla A.2: Conjunto original de fonos del corpus TIMIT.

Apéndice B
El análisis ROC

La representación gráfica de la *curva de características operativas del receptor* (o simplemente *curva ROC*[1]), es una herramienta ampliamente utilizada para visualización, organización y selección de clasificadores basándose en su rendimiento. Las curvas ROC han sido utilizadas en la teoría de detección de señales para representar el equilibrio entre las tasas de aciertos y falsas alarmas en los clasificadores [Egan, 1975]. El análisis de sistemas de detección mediante la curva ROC o simplemente *análisis ROC* se ha ampliado para su uso en la visualización y el análisis del comportamiento de los sistemas de diagnóstico [Swets, 1998].

Uno de los primeros en adoptar el uso de la curva ROC para el aprendizaje automático fue Spackman [1989], quien demostró la utilidad de las curvas ROC en la evaluación y comparación de algoritmos de aprendizaje. En años recientes se ha visto un aumento en el uso de la curva ROC en las comunidades de investigación del aprendizaje automático, debido en parte a la comprensión de que la simple medida de precisión de la clasificación a menudo es un indicador pobre para medir el desempeño de un algoritmo [Provost y Fawcett, 2001].

Para el caso de la detección de palabras clave y considerando el problema de clasificación usando solamente dos clases (*presencia* o *ausencia* de la palabra clave en la frase pronunciada), formalmente decimos que cada instancia I se asigna a un elemento del conjunto $\{\mathbf{p}, \mathbf{n}\}$ de etiquetas de clase *positivo* y *negativo*. Nuestro modelo de clasificación (o clasificador) realiza

[1]Por sus siglas del inglés Receiver Operating Characteristics.

una asignación o mapeo de las instancias a las clases predichas. Produce una salida continua (la estimación de una probabilidad de pertenencia o no a la instancia de una clase) a la cual se le pueden aplicar diferentes umbrales para predecir su pertenencia a una de las clases. Otros modelos producen una etiqueta de clase discreta que indica sólo la clase predicha de la instancia. Para distinguir entre la *clase real* y la *clase predicha* usamos las etiquetas $\{\mathcal{R}, \mathcal{P}\}$ para las predicciones de la clase producidas por el modelo.

Dado un clasificador y una instancia o ejemplo, hay cuatro posibles resultados. Si el ejemplo es positivo y se clasifica como positivo, se cuenta como un *verdadero positivo*, si éste se clasifica como negativo, se cuenta como un *falso negativo*. Si el ejemplo es negativo y se clasifica como negativo, se cuenta como un *verdadero negativo* y si éste último se clasifica como positivo, se cuenta como un *falso positivo*. Dado un clasificador y un conjunto de instancias (el conjunto de pruebas), se puede construir una matriz de confusión de dos por dos (también denominada tabla de contingencia) para representar las disposiciones del conjunto de instancias. Esta matriz es la base de varias medidas de uso común.

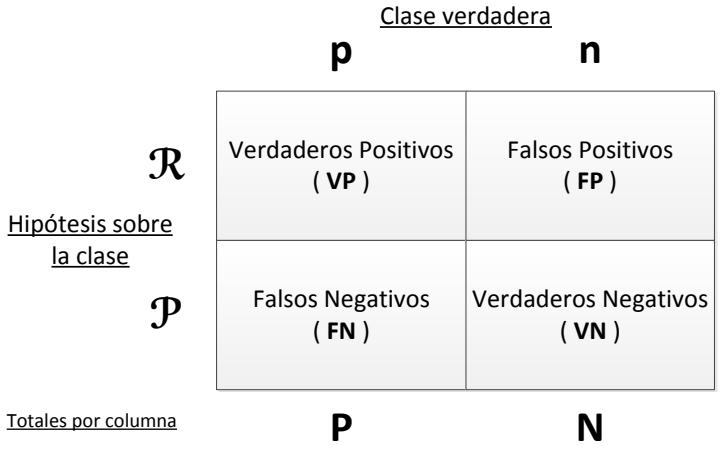

FIGURA B.1: Matriz de confusión.

La Fig. B.1 muestra la matriz de confusión y a continuación se presentan las ecuaciones de varias medidas comúnes que pueden calcularse a partir de

la misma. En la matriz de confusión los números a lo largo de la diagonal principal representan las decisiones hechas correctamente y los números en la diagonal opuesta representan los errores (la confusión) entre las distintas clases. La *tasa de verdaderos positivos* (también llamada *tasa de aciertos*) de un clasificador se calcula como

$$\textit{Tasa de vp} = \text{Sensibilidad} \approx \frac{\text{Positivos correctamente clasificados}}{\text{Total de positivos}} = \frac{TP}{P}$$

La *tasa de falsos positivos* (también llamada *tasa de falsas alarmas*) de un clasificador es

$$\textit{Tasa de fp} \approx \frac{\text{Negativos incorrectamente clasificados}}{\text{Total de negativos}} = \frac{FP}{N}$$

Algunos otros términos asociados a la curva ROC son:

$$\textit{Especificidad} = \frac{\text{Verdaderos negativos}}{\text{Falsos positivos} \ + \ \text{Verdaderos negativos}} = 1 - \textit{Tasa de fp}$$

$$\textit{Valor predictivo positivo} = \text{Precisión} = \frac{TP}{TP + FP}$$

$$\textit{Exactitud} = \frac{TP + TN}{P + N}$$

La curva ROC se representa mediante una gráfica bidimensional en la que se traza la *tasa de vp* en el eje Y y la *tasa de fp* en el eje X. Esta gráfica representa las ventajas y desventajas relativas entre los beneficios (verdaderos positivos) y los costos (falsos positivos). La Fig. B.2 muestra una gráfica de la curva ROC para cinco clasificadores discretos[2] etiquetados de la **A** a la **E**. Cada clasificador discreto produce un par (*tasa de fp*, *tasa de vp*) que corresponde a un solo punto en el espacio ROC.

Hay varios puntos en el espacio ROC que son importantes para tener en cuenta. El punto inferior izquierdo $(0, 0)$ representa la estrategia de no emitir nunca una clasificación positiva, como por ejemplo en un clasificador que no

[2]Un clasificador *discreto* es aquel que emite sólo una etiqueta de clase.

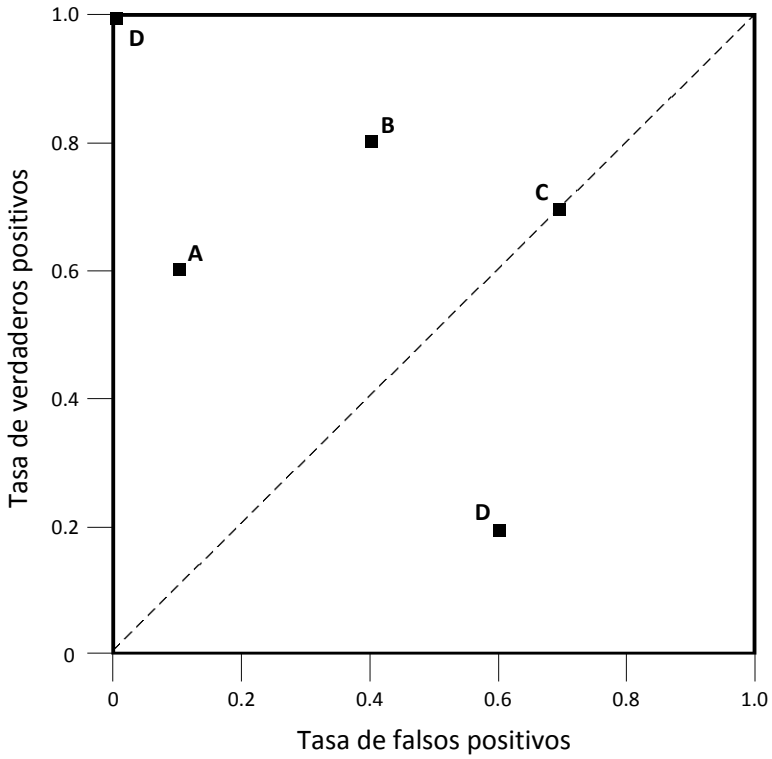

FIGURA B.2: Representación gráfica de la curva ROC mostrando cinco clasificadores discretos.

comete ningún error emitiendo falsos positivos. Es decir para nuestro clasificador corresponde al detector que nunca declara que la palabra está presente y por lo tanto crea una probabilidad nula para ambos eventos: detección o falsa aceptación. La estrategia opuesta, es aquella en donde de manera incondicional se emiten clasificaciones positivas, y está representada por el punto superior derecho $(1, 1)$. Es decir el punto que corresponde a declarar que la palabra siempre está presente. El punto $(0, 1)$ representa la clasificación perfecta.

El rendimiento del clasificador **D** que se muestra en la Fig. B.2 es perfecto. Informalmente decimos que un punto en el espacio ROC es mejor que

otro si se encuentra ubicado hacia el noroeste del primero (la *tasa de vp* es mayor, la *tasa de fp* es inferior o ambas). Los clasificadores que aparecen en el lado izquierdo de una gráfica de la curva ROC, cerca del eje X, pueden ser considerados como "conservadores" ya que clasifican positivamente sólo si existe una fuerte evidencia para ello, de forma que tienen pocos errores de falsos positivos pero a menudo tienen baja tasas de verdaderos positivos. Por ejemplo en la Fig. B.2 el clasificador A es más conservador que el $\mathbf{B}$[3].

La recta de 45 grados en donde $y = x$, representa la estrategia de adivinar o clasificar al azar la pertenencia de una instancia a una clase. Por ejemplo, si un clasificador supone al azar la clase positiva la mitad del tiempo, se puede esperar obtener la mitad de positivos y la mitad de negativos correctos; esto produce el punto $(0,5, 0,5)$ en el espacio ROC. Si se hace una suposición de la clase positiva el 90 % del tiempo, se puede esperar obtener 90 % de los positivos correctos pero su tasa de falsos positivos también aumentará a 90 %, entregando así el punto $(0,9, 0,9)$ en el espacio ROC.

Así, una clasificación al azar producirá un punto de la curva ROC que "se desliza" de ida y vuelta en la diagonal basándose en la frecuencia con que se supone la clase positiva. Con la finalidad de alejarse de esta diagonal a la región triangular superior en el espacio ROC, el clasificador debe aprovechar alguna información en los datos de entrada. En la Fig. B.2, el rendimiento del clasificador $\mathbf{C}$ es prácticamente al azar. En el punto $(0,7, 0,7)$, puede decirse que este clasificador (C) supone la clase positiva el 70 % del tiempo.

También se puede decir que cualquier clasificador que aparece en el triángulo de abajo a la derecha se desempeña peor que si se hiciera la clasificación aleatoriamente, por lo tanto este triángulo suele estar vacío en las gráficos de la curva ROC. La decisión de cualquier clasificador que produce un punto en el triángulo inferior derecho puede ser negada para producir un punto en el triángulo superior izquierdo. Es decir negando cada una de las salidas del clasificador, revirtiendo sus decisiones en cada instancia, convirtiendo la clasificación de verdaderos positivos en errores de falsos negativos y sus falsos positivos en verdaderos negativos.

[3]En la práctica, los clasificadores con un gran número de casos negativos son predominates, por lo que el rendimiento en el extremo izquierdo de la gráfica de la curva ROC se vuelve más interesante.

En la Fig. B.2, el clasificador **E** realiza su trabajo mucho peor que si tomara decisiones al azar y de hecho es la negación de la salida del clasificador **B**. De lo anterior podemos decir que cualquier clasificador que se ubique en la diagonal no toma en cuenta la información acerca de la clase. Y que un clasificador que se ubica debajo de la diagonal toma en cuenta información útil sobre la clase, pero está utilizando la información incorrectamente [Flach y Wu, 2005].

Dado el conjunto de datos a la salida de nuestro clasificador, deseamos construir la curva ROC a partir de ellos y para esto se puede aprovechar la propiedad de monotonicidad de las *clasificaciones con umbral*, esto es, aprovechar el hecho de que cualquier instancia que se clasifica positivamente con respecto a un determinado umbral también se clasifica positivamente para todos los umbrales inferiores. Por consiguiente, podemos comenzar ordenando en forma descendente las instancias de prueba tomando en cuenta solo el valor de sus puntuaciones f e ir recorriendo la lista hacia abajo, procesando una a la vez cada ejemplo e ir actualizando los valores de los verdaderos positivos (VP) y falsos positivos (FP) a medida que avanzamos. De esta manera se puede ir construyendo la gráfica de la curva ROC a partir de una exploración lineal.

El algoritmo para generar la curva ROC se muestra a en el *Algoritmo 1*. Comenzamos igualando los valores de VP y FP a cero. Para cada ejemplo positivo incrementamos VP y para cada caso negativo incrementamos FP. Mientras tanto, es necesario mantener una pila *(R)* de puntos de la curva ROC, empujando hacia abajo cada nuevo punto en R después de cada instancia procesada. Obteniendo como resultado final de este proceso la pila R, la cual contendrá los pares de puntos que definen la curva ROC.

Hasta aquí queda claro que una curva ROC es una representación bidimensional del desempeño de un clasificador. Pero con el objetivo de comparar dos clasificadores lo que quisiéramos es poder reducir la información acerca de dicha medida de desempeño (contenida en la curva ROC), a un valor escalar que represente el rendimiento esperado. Para ello un método común consiste en calcular el *área bajo la curva* ROC (o de forma abreviada ABC) [Bradley, 1997]. Dado que el ABC es una porción del área de un cuadrado con lados de valor iguales a la unidad, su valor siempre estará entre 0 y 1. Sin embargo y dado que el resultado de suponer aleatoriamente produce la

Algoritmo 4 Algoritmo para encontrar los puntos de la curva ROC

Entrada: El conjunto de datos de prueba L. La estimación $f(i)$ del clasificador probabilístico de que el ejemplo i es positivo. El número total de ejemplos positivos P y negativos N.

Salida: Lista R de puntos de la curva ROC para diferentes *tasas de falsos positivos* (TFP).

Requiere: $P > 0$ y $N > 0$

1: $L_{ordenado} \longleftarrow L$ ordenado en forma decreciente de acuerdo a los puntajes dados por f.
2: FP $\longleftarrow$ VP $\longleftarrow 0$
3: $R \longleftarrow \langle\rangle$
4: $f_{previo} \longleftarrow -\infty$
5: $i \longleftarrow 1$
6: **Mientras** $i \leq |L_{ordenado}|$ **hacer**
7: **Si** $f(i) \neq f_{previo}$ **entonces**
8: Poner $\left(\frac{FP}{N}, \frac{VP}{P}\right)$ en R.
9: $f_{previo} \longleftarrow f(i)$
10: **fin Si**
11: **Si** $L_{ordenado}[i]$ es un ejemplo positivo **entonces**
12: $VP \longleftarrow VP + 1$
13: **si no**
14: $FP \longleftarrow FP + 1$
15: **fin Si**
16: $i \longleftarrow i + 1$
17: **fin Mientras**
18: Poner $\left(\frac{FP}{N}, \frac{VP}{P}\right)$ en R. /* Se trata de $(1, 1)$ */
19: **Fin**

línea diagonal entre los puntos (0,0) y (1, 1), que implica una área de 0,5, ningún clasificador realista debería tener un valor de ABC menor a 0,5.

El ABC tiene una propiedad estadística importante que es equivalente a la probabilidad de que el clasificador otorgue un puntaje más alto a una instancia positiva elegida al azar que a una instancia negativa, también elegida al azar. A pesar de esto, en la práctica el uso del valor del ABC tiene un muy buen desempeño y a menudo se utiliza como medida general de la capacidad predictiva deseable.

El ABC puede calcularse fácilmente haciendo una modificación menor al procedimiento para obtener los puntos que definen a la curva ROC (descrito anteriormente), como se muestra en el algoritmo para el cálculo del ABC que se presenta en el *Algoritmo 5*. En lugar de recolectar los puntos de la curva ROC, el algoritmo añade sucesivamente valores de áreas de trapezoides al valor total del ABC[4]. Los trapezoides se utilizan en lugar de rectángulos con el fin de promediar el efecto entre cada punto de la curva. Por último, el procedimiento divide el área bajo la curva entre el área total posible para escalar el valor al cuadrado con lados iguales a la unidad.

A continuación presentamos las curvas ROC obtenidas a partir de los datos en la salida de los clasificadores discriminatorio y basado en MOMs, señalando en cada gráfica el punto óptimo de operación obtenido, que asegura la mejor relación posible entre sensibilidad y especificidad. Es decir entre las proporciones de casos positivos (TPV) y de casos negativos (TNV) que fueron correctamente identificados.

En la Fig. B.5 se describen las matrices de confusión obtenidas para los clasificadores discriminatorio y basado en MOMs en el punto óptimo de operación y en la Tabla B.1 se muestran los valores de sensibilidad y especificidad de ambos clasificadores para dichos puntos de operación.

[4]El cálculo de los trapezoides se realiza con ayuda de la función AREA_TRAPEZOIDE que se describe en el *Algoritmo 6*.

Algoritmo 5 Algoritmo para calcular el área bajo la curva ROC

Entrada: El conjunto de datos de prueba L. La estimación $f(i)$ del clasificador probabilístico de que el ejemplo i es positivo. El número total de ejemplos positivos P y negativos N.

Salida: El área A bajo la curva ROC.

Requiere: $P > 0$ y $N > 0$

1: $L_{ordenado} \longleftarrow L$ ordenado en forma decreciente de acuerdo a los puntajes dados por f.
2: $\text{FP} \longleftarrow \text{VP} \longleftarrow 0$
3: $\text{FP}_{previo} \longleftarrow \text{VP}_{previo} \longleftarrow 0$
4: $A \longleftarrow 0$
5: $f_{previo} \longleftarrow -\infty$
6: $i \longleftarrow 1$
7: **Mientras** $i \leq |L_{ordenado}|$ **hacer**
8: **Si** $f(i) \neq f_{previo}$ **entonces**
9: $A \longleftarrow A+$ AREA_TRAPEZOIDE(FP,FP_{previo},VP,VP_{previo})
10: $f_{previo} \longleftarrow f(i)$
11: $\text{FP}_{previo} \longleftarrow \text{FP}$
12: $\text{VP}_{previo} \longleftarrow \text{VP}$
13: **fin Si**
14: **Si** i es un ejemplo positivo **entonces**
15: $VP \longleftarrow VP + 1$
16: **si no**
17: $FP \longleftarrow FP + 1$
18: **fin Si**
19: $i \longleftarrow i + 1$
20: **fin Mientras**
21: $A \longleftarrow A+$ AREA_TRAPEZOIDE(N,FP_{previo},N,VP_{previo})
22: $A \longleftarrow \dfrac{A}{P \times N}$
23: **Fin**

Algoritmo 6 Función utilizada para el cálculo del área bajo la curva ROC, que calcula el área de un trapezoide.

1: **Función** AREA_TRAPEZOIDE$(X1, X2, Y1, Y2)$
2: Base $\longleftarrow |X1 - X2|$
3: Altura$_{promedio} \longleftarrow \dfrac{(Y1 + Y2)}{2}$
4: **Regresa** Base $\times$ Altura$_{promedio}$
5: **Fin Función**

Clasificador	**Pto. operación**	**Sensibilidad**	**Especificidad**
Discriminatorio	0.3620	0.9667	0.9250
MOM	0.4676	0.8182	0.9333

TABLA B.1: Valores de sensibilidad (*tvp*) y especificidad (1-*tfp*) para los clasificadores discriminatorio y basado en MOMs, en sus puntos óptimos de operación.

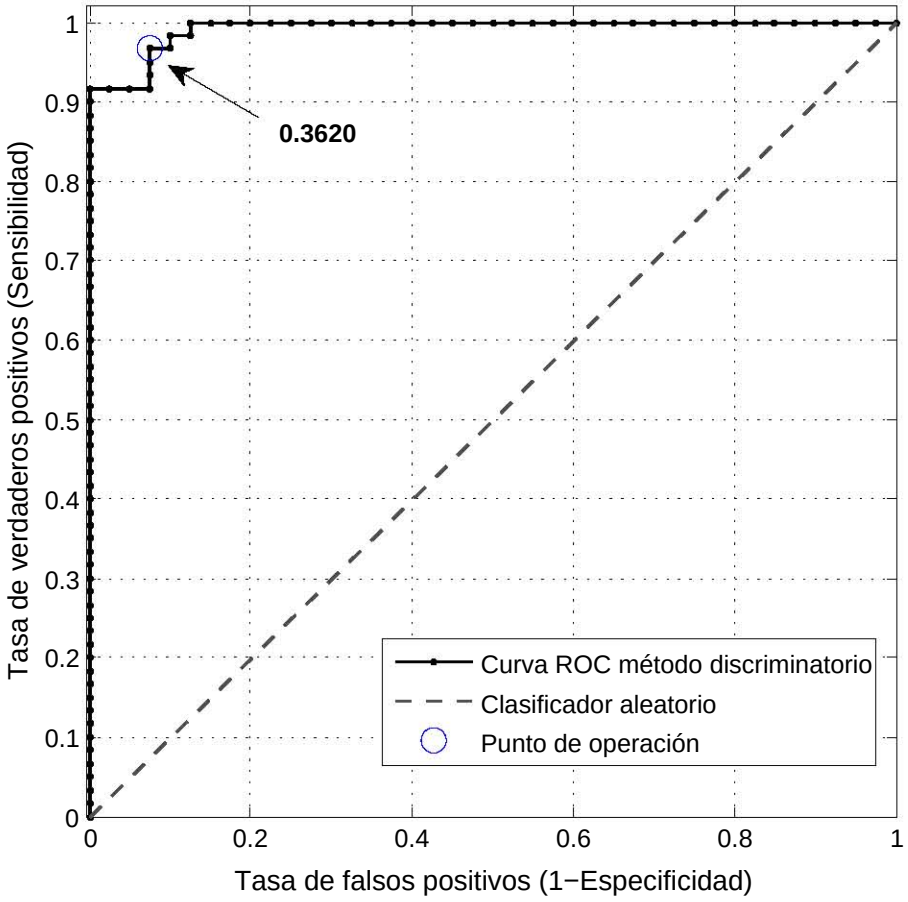

Figura B.3: Curva ROC para el clasificador discriminatorio señalando el punto de operación que ofrece la mejor relación entre sensibilidad y especificidad.

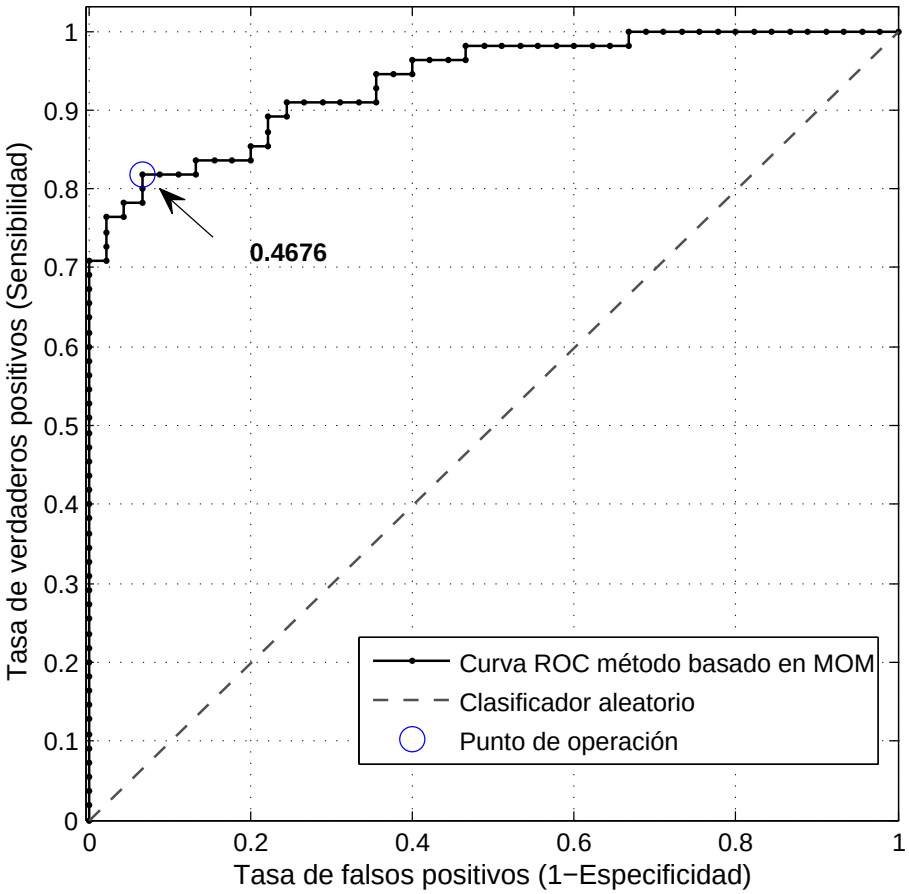

FIGURA B.4: Curva ROC para el clasificador basado en MOM señalando el punto de operación que ofrece la mejor relación entre sensibilidad y especificidad.

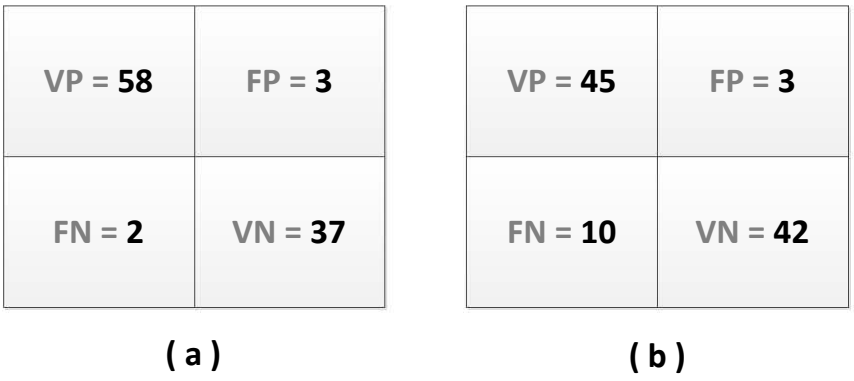

Figura B.5: Matrices de confusión para **(a)** el clasificador basado en el modelo discriminatorio propuesto, en su punto óptimo de operación **0.3620** y **(b)** el clasificador basado en MOMs en su punto de óptimo de operación 0.4676.

Bibliografía

Allen, J. B. *Articulation and Intelligibility*. Synthesis Lectures on Speech and Audio Processing. Morgan & Claypool Publishers, 2005.

Baghai-Ravary, L. Evidence for the strength of the relationship between automatic speech recognition and phoneme alignment performance. En *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, páginas 5262 –5265. 2010.

Bahl, L. R., Brown, P. F., de Souza, P. V. y Mercer, R. L. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. En *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, páginas 49–52. Tokyo, 1986.

Baker, J., Deng, L., Khudanpur, S., Lee, C.-H., Glass, J., Morgan, N. y O'Shaughnessy, D. Updated minds report on speech recognition and understanding, part 2 [dsp education]. *Signal Processing Magazine, IEEE*, vol. 26(4), páginas 78 –85, 2009.

Bartlett, P. y Taylor, S. J. Generalization Performance of Support Vector Machines and Other Pattern Classifiers. En *Advances in Kernel Methods — Support Vector Learning* (editado por B. Schölkopf, C. J. C. Burges y A. J. Smola), páginas 43–54. MIT Press, Cambridge, MA, 1999.

Baum, L. E., Petrie, T., Soules, G. y Weiss, N. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics*, vol. 41(1), páginas 164–171, 1970.

BENAYED, Y., FOHR, D., HATON, J. y CHOLLET, G. Confidence measures for keyword spotting using support vector machines. En *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, vol. 1, páginas I–588 – I–591 vol.1. 2003.

BENESTY, J., SONDHI, M. M. y HUANG, Y., editores. *Springer Handbook of Speech Processing*. Springer, Berlin, 2008.

BOURLARD, H. Towards increasing speech recognition error rates. En *EUROSPEECH*. ISCA, 1995.

BRADLEY, A. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, vol. 30(7), páginas 1145–1159, 1997.

BRIDLE, J. An efficient elastic-template method for detecting given words in running speech. En *Proc. of the British Acoustic Society Meeting,*, vol. 84, páginas 1–4. 1973.

BRUGNARA, F., FALAVIGNA, D. y OMOLOGO, M. Automatic segmentation and labeling of speech based on hidden markov models. *Speech Communication*, vol. 12(4), páginas 357–370, 1993.

BURGES, C. J. C. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, vol. 2(2), páginas 121–167, 1998.

CESA-BIANCHI, N., CONCONI, A. y GENTILE, C. On the Generalization Ability of On-Line Learning Algorithms. *IEEE Transactions on Information Theory*, vol. 50(9), páginas 2050–2057, 2004.

CHANG, E. I.-C. *Improving wordspotting performance with limited training data*. Tesis Doctoral, Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.

CLARKSON, P. y MORENO, P. J. On the use of support vector machines for phonetic classification. En *in ICASSP99*, páginas 585–588. 1999.

CORTES, C. y MOHRI, M. Confidence intervals for the area under the roc curve. En *NIPS*. 2004.

CORTES, C. y VAPNIK, V. Support-vector networks. *Machine Learning*, vol. 20(3), páginas 273–297, 1995.

COSI, P., FALAVIGNA, D. y OMOLOGO, M. A preliminary statistical evaluation of manual and automatic segmentation discrepancies. En *EUROSPEECH*. ISCA, 1991.

COX, S., BRADY, R. y JACKSON, P. Techniques for accurate automatic annotation of speech waveforms. En *ICSLP*. ISCA, 1998.

CRAMMER, K., DEKEL, O., KESHET, J., SHALEV-SHWARTZ, S. y SINGER, Y. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, vol. 7, páginas 551–585, 2006.

CRISTIANINI, N. y SHAWE-TAYLOR, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge, UK, 2000. ISBN 0521780195.

DEKEL, O., KESHET, J. y SINGER, Y. An online algorithm for hierarchical phoneme classification. En *MLMI* (editado por S. Bengio y H. Bourlard), vol. 3361 de *Lecture Notes in Computer Science*, páginas 146–158. Springer, 2004.

DELLER, J. J. R., HANSEN, J. H. L. y PROAKIS, J. G. *Discrete-Time Processing of Speech Signals*. Wiley-IEEE Press, 2da edición, 1999.

DEMPSTER, A. P., LAIRD, N. M. y RUBIN, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B*, vol. 39, 1977.

DIGALAKIS, V., OSTENDORF, M. y ROHLICEK, J. Fast algorithms for phone classification and recognition using segment-based models. *Signal Processing, IEEE Transactions on*, vol. 40(12), páginas 2885 –2896, 1992.

EGAN, J. P. *Signal detection theory and ROC analysis*. Series in Cognition and Perception. Academic Press, New York, NY, 1975.

ES-202-211. ETSI standars for Distributed Speech Recognition. European Telecommunications Standards Institute. Extended front-end feature extraction algorithm, 2003. [en línea] `http://www.etsi.org/website/technologies/distributedspeechrecognition.aspx`.

FARHAT, A., PERENNOU, G. y ANDRÉ-OBRECHT, R. A segmental approach versus a centisecond one for automatic phonetic time-alignment. En *EUROSPEECH*. ISCA, 1993.

FINK, M., SHALEV-SHWARTZ, S., SINGER, Y. y ULLMAN, S. Online multiclass learning by interclass hypothesis sharing. En *Proceedings of the 23rd international conference on Machine learning*, páginas 313–320. ACM, Pittsburgh, Pennsylvania, 2006.

FLACH, P. A. y WU, S. Repairing concavities in roc curves. En *IJCAI* (editado por L. P. Kaelbling y A. Saffiotti), páginas 702–707. Professional Book Center, 2005.

FUJIHARA, H. y GOTO, M. Three techniques for improving automatic synchronization between music and lyrics: Fricative detection, filler model, and novel feature vectors for vocal activity detection. En *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, páginas 69 –72. 2008.

FURUI, S. 50 years of progress in speech and speaker recognition research. *ECTI Transactions on Computer and Information Technology*, vol. 1(2), páginas 64–74, 2005.

GAROFOLO, J. S., LAMEL, L. F., FISHER, W. M., FISCUS, J. G., PALLETT, D. S. y DAHLGREN, N. L. DARPA TIMIT acoustic phonetic continuous speech corpus CDROM. 1993.

GHOLAMPOUR, I. y NAYEBI, K. A new fast algorithm for automatic segmentation of continuous speech. En *ICSLP*. ISCA, 1998.

GRANGIER, D. y BENGIO, S. A discriminative kernel-based approach to rank images from text queries. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30(8), páginas 1371 –1384, 2008.

GUBRYNOWICZ, R. y WRZOSKOWICZ, A. Labeller - a system for automatic labelling of speech continuous signal. En *EUROSPEECH*. ISCA, 1993.

HATAZAKI, K., KOMORI, Y., KAWABATA, T. y SHIKANO, K. Phoneme segmentation using spectrogram reading knowledge. En *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, páginas 393 –396 vol.1. 1989.

HERBSTER, M. Learning additive models online with fast evaluating kernels. En *COLT/EuroCOLT* (editado por D. P. Helmbold y B. Williamson), vol.

2111 de *Lecture Notes in Computer Science*, páginas 444–460. Springer, 2001.

HIGGINS, A. y WOHLFORD, R. Keyword recognition using template concatenation. En *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '85.*, vol. 10, páginas 1233 – 1236. 1985.

HOSOM, J.-P. Automatic phoneme alignment based on acoustic-phonetic modeling. En *INTERSPEECH* (editado por J. H. L. Hansen y B. L. Pellom). ISCA, 2002.

HOSOM, J.-P. Speaker-independent phoneme alignment using transition-dependent states. *Speech Communication*, vol. 51(4), páginas 352–368, 2009.

HOUBEN, C. G. J. Automatic labelling of speech using an acoustic-phonetic knowledge base. En *EUROSPEECH*, páginas 2104–2107. 1989.

HTK. HTK ver. 3.4.1. Hidden Markov Model Toolkit. Cambridge University Engineering Dept. (CUED)., 2009. [en línea] `http://htk.eng.cam.ac.uk/`.

HUGGINS-DAINES, D. y RUDNICKY, A. I. A constrained baum-welch algorithm for improved phoneme segmentation and efficient training. En *INTERSPEECH*. ISCA, 2006.

JELINEK, F. *Statistical Methods for Speech Recognition (Language, Speech, and Communication)*. The MIT Press, 1998.

JUANG, B.-H. y KATAGIRI, S. Discriminative learning for minimum error classification [pattern recognition]. *Signal Processing, IEEE Transactions on*, vol. 40(12), páginas 3043 –3054, 1992.

JUNKAWITSCH, J., RUSKE, G. y HÖGE, H. Efficient methods for detecting keywords in continuous speech. En *EUROSPEECH* (editado por G. Kokkinakis, N. Fakotakis y E. Dermatas). ISCA, 1997.

JUNQUA, J.-C. y HATON, J.-P. *Robustness in Automatic Speech Recognition: Fundamentals and Applications*. Kluwer Academic Publishers, Norwell, MA, USA, 1995.

KABRÉ, H., PERENNOU, G. y VIGOUROUX, N. A non-linear filtering method applied to automatic segmentation of multilingual speech corpora. En *EUROSPEECH*. 1991.

KACUR, J., CEPKO, J. y PALENIK, A. Automatic labeling schemes for concatenative speech synthesis. En *ELMAR, 2008. 50th International Symposium*, vol. 2, páginas 639 –642. 2008.

KARSMAKERS, P., PELCKMANS, K., SUYKENS, J. y HAMME, H. V. Fixed-size kernel logistic regression for phoneme classification. En *INTERSPEECH*. ISCA, 2007.

KATZ, S. M. Estimation of probabilities from sparse data for the language model component of a speech recognizer. En *IEEE Transactions on Acoustics, Speech and Singal processing*, 3, páginas 400–401. 1987.

KAWABATA, T., HANAZAWA, T. y S., K. Word spotting method based on hmm phoneme recognition. En *Journal Acoustic Society of America*, vol. 84. 1988.

KEATING, P. A., BYRD, D., FLEMMING, E. y TODAKA, Y. Phonetic analyses of word and segment variation using the timit corpus of american english. *Speech Communication*, vol. 14(2), páginas 131 – 142, 1994.

KESHET, J., CHAZAN, D., BOBROVSKY, B.-Z. y BOBROVSKY, B.-Z. Plosive spotting with margin classifiers. En *INTERSPEECH*, páginas 1637–1640. 2001.

KIVINEN, J., SMOLA, A. y WILLIAMSON, R. Online learning with kernels. *Signal Processing, IEEE Transactions on*, vol. 52(8), páginas 2165 – 2176, 2004.

KIVINEN, J. y WARMUTH, M. K. Exponentiated gradient versus gradient descent for linear predictors. *Inf. Comput.*, vol. 132(1), páginas 1–63, 1997.

KOLLER, D. y SAHAMI, M. Hierarchically classifying documents using very few words. En *Proceedings of ICML-97, 14th International Conference on Machine Learning* (editado por D. H. Fisher), páginas 170–178. Morgan Kaufmann Publishers, San Francisco, US, Nashville, US, 1997.

LAMEL, L. F., KASSEL, R. y SENEFF, S. Speech database development: Design and analysis of the acoustic-phonetic corpus. En *Proceedings of the DARPA Speech Recognition Workshop*, páginas 100–110. 1986.

LEE, C.-H., CLEMENTS, M. A., DUSAN, S., FOSLER-LUSSIER, E., JOHNSON, K., JUANG, B.-H. y RABINER, L. R. An overview on automatic speech attribute transcription (asat). En *INTERSPEECH*, páginas 1825–1828. ISCA, 2007.

LEE, K. F. y HON, H. W. Speaker-independent phone recognition using hidden Markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37(11), páginas 1641–1648, 1989.

LJOLJE, A. y RILEY, M. D. Automatic segmentation of speech for tts. En *EUROSPEECH*. ISCA, 1993.

MALFRÈRE, F., DEROO, O. y DUTOIT, T. Phonetic alignment: speech synthesis based vs. hybrid hmm/ann. En *ICSLP*. ISCA, 1998.

MANN, H. y WHITNEY, D. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, páginas 50–60, 1947.

MANNING, C. y SCHÜTZE, H. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, 1999.

MATĚJKA, P. *Phonotactic and Acoustic Language Recognition*. Tesis Doctoral, Brno University of Technology, Faculty of Electrical Engineering and Communication, 2008.

MAYNE, A. y PERRY, R. Hierarchically classifying documents with multiple labels. En *CIDM*, páginas 133–139. IEEE, 2009.

MCCALLUM, A., ROSENFELD, R., MITCHELL, T. M. y NG, A. Y. Improving text classification by shrinkage in a hierarchy of classes. En *ICML* (editado por J. W. Shavlik), páginas 359–367. Morgan Kaufmann, 1998.

MITCHELL, C., HARPER, M. y JAMIESON, L. Using explicit segmentation to improve hmm phone recognition. En *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, vol. 1, páginas 229 –232 vol.1. 1995.

MOHAMED, A., DAHL, G. y HINTON, G. Acoustic modeling using deep belief networks. *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. PP(99), página 1, 2011.

MORRIS, J. y FOSLER-LUSSIER, E. Conditional random fields for integrating local discriminative classifiers. *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16(3), páginas 617 –628, 2008.

OSTENDORF, M., DIGALAKIS, V. V. y KIMBALL, O. A. From HMM's to segment models: A unified view of stochastic modeling for speech recognition. *IEEE Trans. on Speech and Audio Proc.*, vol. 4(5), páginas 360–378, 1996.

POVEY, D. y WOODLAND, P. Minimum phone error and i-smoothing for improved discriminative training. En *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 1, páginas I–105 –I–108. 2002.

PROVOST, F. J. y FAWCETT, T. Robust classification for imprecise environments. *Machine Learning*, vol. 42(3), páginas 203–231, 2001.

RABINER, L. y JUANG, B. *Fundamentals of Speech Recognition*. Prentice-Hall Signal Processing Series, Englewood Cliffs, New Jersey, 1993.

RABINER, L. y SCHAFER, R. *Digital Processing of Speech Signals*. Englewood Cliffs: Prentice Hall, 1978.

REYNOLDS, T. J. y ANTONIOU, C. A. Experiments in speech recognition using a modular mlp architecture for acoustic modelling. *Inf. Sci. Inf. Comput. Sci.*, vol. 156, páginas 39–54, 2003. ISSN 0020-0255.

ROSE, R. y MOMAYYEZ, P. Integration of multiple feature sets for reducing ambiguity in asr. En *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4, páginas IV–325 –IV–328. 2007.

ROSE, R. y PAUL, D. A hidden markov model based keyword recognition system. En *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, páginas 129 –132 vol.1. 1990.

SAITO, T. On the use of f0 features in automatic segmentation for speech synthesis. En *ICSLP*. ISCA, 1998.

SALOMON, J. Support vector machines for phoneme classification. *Artificial Intelligence*, 2001.

SALOMON, J., KING, S., SALOMON, J. y SALOMON, J. Framewise phone classification using support vector machines. En *INTERSPEECH*. 2002.

SANDNESS, E. D. y HETHERINGTON, I. L. Keyword-based discriminative training of acoustic models. En *INTERSPEECH*, páginas 135–138. ISCA, 2000.

SCANLON, P., ELLIS, D. P. W. y REILLY, R. B. Using broad phonetic group experts for improved speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15(3), páginas 803 –812, 2007.

SCHÖLKOPF, B. y SMOLA, A. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. MIT Press, Cambridge, MA, 2002.

SHA, F. y SAUL, L. Large margin gaussian mixture modeling for phonetic classification and recognition. En *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1, página I. 2006.

SHALEV-SHWARTZ, S., KESHET, J., SINGER, Y. y SINGER, Y. Learning to align polyphonic music. En *ISMIR*. 2004.

SILAGHI, M.-C., BOURLARD, H. y ECUBLENS, E. Posterior-based keyword spotting approaches without filler models - iterative viterbi decoding and one-pass approaches. En *Proc. of the IEEE Automatic Speech Recognition and Understanding Work-shop*, páginas 213–216. Keystone, 1999.

SINISCALCHI, S., SCHWARZ, P. y LEE, C.-H. High-accuracy phone recognition by combining high-performance lattice generation and knowledge based rescoring. En *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4, páginas IV–869 –IV–872. 2007.

SPACKMAN, K. A. Signal detection theory: Valuable tools for evaluating inductive learning. En *ML*, páginas 160–163. 1989.

SWETS, J. A. Measuring the accuracy of diagnostic systems. *Science*, vol. 240, páginas 1285–1289, 1998.

SZÖKE, I., SCHWARZ, P., MATEJKA, P., BURGET, L., KARAFIÁT, M., FAPSO, M. y CERNOCKÝ, J. Comparison of keyword spotting approaches for informal continuous speech. En *INTERSPEECH*, páginas 633–636. ISCA, 2005.

TASKAR, B., GUESTRIN, C. y KOLLER, D. Max-margin markov networks. En *Advances in neural information processing systems 16: proceedings of the 2003 conference*. 2003. Stanford University.

TOLEDANO, D., GOMEZ, L. y GRANDE, L. Automatic phonetic segmentation. *Speech and Audio Processing, IEEE Transactions on*, vol. 11(6), páginas 617 – 625, 2003.

VAPNIK, V. N. *The nature of statistical learning theory*. Springer, New York, 1995.

VAPNIK, V. N. *Statistical Learning Theory*. Wiley-Interscience, 1998.

VORSTERMANST, A., MARTENS, J.-P., COILE, B. V. y COILE, B. V. Fast automatic segmentation and labeling: results on timit and euromo. En *EUROSPEECH*. 1995.

WEINTRAUB, M. Lvcsr log-likelihood ratio scoring for keyword spotting. En *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, vol. 1, páginas 297 –300 vol.1. 1995.

WEINTRAUB, M., BEAUFAYS, F., RIVLIN, Z., KONIG, Y. y STOLCKE, A. Neural-network based measures of confidence for word recognition. En *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 2, páginas 887 –890 vol.2. 1997.

WESTON, J. y WATKINS, C. Support vector machines for multi-class pattern recognition. En *ESANN*, páginas 219–224. 1999.

WILCOXON, F. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, vol. 1(6), páginas 80–83, 1945. ISSN 00994987.

WILPON, J., RABINER, L., LEE, C.-H. y GOLDMAN, E. Automatic recognition of keywords in unconstrained speech using hidden markov models. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 38(11), páginas 1870 –1878, 1990.

YARRINGTON, D., BUNNELL, H. T. y BALL, G. Robust automatic extraction of diphones with variable boundaries. En *EUROSPEECH*. ISCA, 1995.

YOUNG, S. A review of large-vocabulary continuous-speech recognition. *Signal Processing Magazine, IEEE*, vol. 13(5), páginas 45–57, 1996.

YOUNG, S. J., KERSHAW, D., ODELL, J., OLLASON, D., VALTCHEV, V. y WOODLAND, P. *The HTK Book Version 3.4*. Cambridge University Press, 2006.

ZUE, V., SENEFF, S. y GLASS, J. Speech database development at mit: Timit and beyond. *Speech Communication*, vol. 9(4), páginas 351 – 356, 1990.

ZUE, V. W. y SENEFF, S. - transcription and alignment of the timit database. En *Recent Research Towards Advanced Man-Machine Interface Through Spoken Language* (editado por H. Fujisaki), páginas 515 – 525. Elsevier Science B.V., Amsterdam, 1996.