Reducción de espacios de entrenamiento empleando modelos ocultos de Markov basados en entrenamiento discriminativo

Julián David Arias Londoño

Universidad Nacional de Colombia
Facultad de Ingeniería y Arquitectura
Maestría en Ingeniería - Automatización Industrial
Manizales
2007

Reducción de espacios de entrenamiento empleando modelos ocultos de Markov basados en entrenamiento discriminativo

Julián David Arias Londoño

Trabajo de grado para optar al título de
Magíster en Ingeniería — Automatización Industrial

Director
Ph.D. César Germán Castellanos Domínguez

Universidad Nacional de Colombia
Faculty of Engineering and Architecture
Department of Electrical, Electronic and Computing Engineering
Manizales
2007

Reduction of Training Spaces using MCE - based Hidden Markov Models

Julián David Arias Londoño

Thesis for the degree of
Master in Engineering — Industrial Automation

Supervisor
Ph.D. César Germán Castellanos Domínguez

National University of Colombia
Faculty of Engineering and Architecture
Department of Electrical, Electronic and Computing Engineering
Manizales
2007

Este trabajo se realiza en el marco de los proyectos: "*Análisis de variabilidad estocástica en la detección de patologías sobre registros de voz y ECG*", n° $AL06\_EX_PID\_033$, financiado por la Universidad Politécnica de Madrid, en la convocatoria de proyectos de I+D de cooperación con Iberoamérica correspondiente al año 2006. E "*Identificación automatizada de Hipernasalidad en niños con LPH por medio de análisis acústico del habla*", n° 20201004208, financiado por la Dirección de Investigaciones DIMA, de la Universidad Nacional de Colombia Sede Manizales

Índice

Lista de tablas

Lista de figuras

Resumen

Es común en el reconocimiento de patrones que los mayores esfuerzos se realicen en las etapas de medición-extracción de características y de clasificación. En diversos problemas de reconocimiento se encuentra que los parámetros resultantes de la medición de variables presentan una dinámica temporal y que esta dinámica en sí misma, es la que contiene mayor parte de la información discriminante. Las técnicas típicamente utilizadas en la etapa de extracción de características, están diseñadas para variables estáticas, es decir, variables que no presentan ningún tipo de dinámica. Este es el caso de técnicas como PCA y LDA. Surge entonces la necesidad de generar metodologías de extracción de características que tengan en cuenta la información dinámica de las variables. Por otro lado, es conocido que los criterios utilizados en las técnicas de extracción de características difieren del criterio de encontrar mínimo error de clasificación; este hecho genera incompatibilidad entre el criterio utilizado en la etapa de extracción de características y la etapa de clasificación y puede degradar el desempeño del sistema. Se presenta por lo tanto una metodología de diseño simultáneo de una etapa de extracción de características y un clasificador basado en *modelos ocultos de Markov - HMM*, por medio del algoritmo de *mínimo error de clasificación - MCE*. La extracción de características es dependiente de los estados del modelo y es optimizada utilizando el mismo criterio de ajuste de parámetros del HMM. La metodología es validada sobre un problema de reconocimiento de patologías de voz. Los resultados muestran que el entrenamiento de HMM por medio del algoritmo MCE mejora el reconocimiento en comparación con el método de entrenamiento clásico por el criterio de máxima verosimilitud. Además, la metodología propuesta disminuye la similitud entre modelos de clases diferentes y mejora el desempeño del sistema.

Abstract

In pattern recognition is often common that the most of the attention is centered in the measure-extraction and classification stages. In several recognition problems, the obtained measures display a time-variant dynamic and this one contains a high level of the discriminant information. The classical techniques for feature extraction are designed for static features. PCA and LDA are examples of this. At this point becomes necessary the development of dynamic feature extraction methodologies. On the other hand, it is well known that the classical features extraction techniques make use of optimization criteria that are different from the classifier's minimum classification error criterion. This fact may cause inconsistency between feature extraction and the classification stages and consequently, degrade the performance of systems. For all this reasons, a *hidden Markov models (HMM)* - based methodology for simultaneous desing of extraction and classification stages is presented. Such a methodology is based on the *minimum classification error (MCE)* algorithm. The feature extraction is model state - dependent and is optimized using the same criterion of parameter estimation of the HMM. Validation is carried out over a automatic detection of pathological voices problem. The result shows that the MCE training improves the accuracy against the classical maximum likelihood training. In addition, the proposed methodology diminished the similarity between models of different classes and improves the performance systems.

Agradecimientos

Quiero agradecer en primer lugar al profesor Ph.D. Germán Castellanos Domínguez por su constante apoyo a lo largo de la realización de este trabajo, además de su acompañamiento durante todo el desarrollo del postgrado. Agradezco a los Ingenieros: Jorge Alberto Jaramillo Garzón, Genaro Daza Santacoloma, Luis Gonzalo Sánchez, y a el Matemático Fernando Martínez Tabares, integrantes del grupo de investigación Control y Procesamiento Digital de Señales, de la Universidad Nacional de Colombia sede Manizales, por su disposición para participar en las diferentes discusiones, que derivaron en importantes aportes al trabajo. De igual forma, quiero agradecer al MSc. Mauricio A. Álvarez López, profesor de la Universidad Tecnológica de Pereira, porque con sus ideas dio luces importantes al desarrollo del trabajo.
Finalmente, quiero agradecer al profesor Ph.D. Juan Ignacio Godino Llorente y a los doctorandos: Nicolás Sáenz Lechón y Víctor Osma Ruiz, integrantes del grupo de investigación Bioingeniería y Optoelectrónica, perteneciente a la E.U.I.T. Telecomunicación, de la Universidad Politécnica de Madrid, por sus valiosos aportes y por el acompañamiento que permitió la utilización en este trabajo de metodologías desarrolladas en sus investigaciones.

Parte I

Preliminares

Introducción

El objetivo principal de los sistemas de reconocimiento de patrones es clasificar datos de entrada dentro de un número definido de clases. Convencionalmente los sistemas de reconocimiento de patrones tiene dos componentes: análisis de características y clasificación de patrones [1]. El análisis de características se alcanza en dos pasos: extracción de parámetros (*EP*) y extracción de características (*EC*). Típicamente, la etapa de EC se emplea para reducir el tamaño del espacio de representación de los datos, proporcionado en la etapa de EP, de tal manera que se utilicen únicamente las variables que mayor información aportan al proceso de clasificación. Aunque teóricamente la etapa de EC puede no ser necesaria si la etapa de EP diseñada, proporciona un espacio de representación de baja dimensión y alta discriminancia, en la práctica, la etapa de EC se emplea a menudo por problemas de alta dimensionalidad. Utilizar un espacio de representación de alta dimensión presenta principalmente dos problemas: en primer lugar, es conocido que el clasificador más simple requiere que el número de observaciones sea una función exponencial de la dimensión del espacio de características (lo que se conoce como el problema de la dimensionalidad) [2]. Este problema se enfatiza, debido a que diversos problemas de reconocimiento de patrones deben ser abordados en condiciones de baja estadística (poco número de muestras). En segundo lugar, porque reduciendo la dimensionalidad del espacio de características se disminuye la complejidad del clasificador [3].

En diversos problemas de reconocimiento de patrones, se encuentra que los parámetros resultantes de la medición de variables, son parámetros que presentan una dinámica temporal y que esta dinámica es la que contiene mayor parte de la información discriminante que debe ser utilizada por el sistema; este es el caso del problema de detección automática de patologías de voz, en el cual los parámetros comúnmente utilizados, presentan un comportamiento dinámico en el tiempo y su dinámica es altamente relevante en la valoración médica de los pacientes [4]. Las técnicas típicamente utilizadas en la etapa de EC, están diseñadas para variables estáticas, es decir, variables que no presentan ningún tipo de dinámica (este es el caso de técnicas como el *análisis de componentes principales-PCA* y el *análisis discriminante lineal - LDA*), lo que impide la utilización de este tipo de técnicas sobre características dinámicas. Simultáneamente, es conocido que las técnicas clásicamente utilizadas para realizar EC, tienen la desventaja de que su criterio de optimización es diferente al criterio de diseño de la etapa de clasificación, que consiste en obtener mínimo error de reconocimiento [1]. Este hecho puede causar inconsistencia entre las etapas de EC y clasificación de un sistema de reconocimiento de patrones y consecuentemente, degradar el desempeño del sistema [5].

Por otro lado, los modelos ocultos de Markov (*Hidden Markov Models (HMM)*) son una

clase de procesos estocásticos que permiten modelar series de tiempo y han sido empleadas en el procesamiento de secuencias de datos temporales aplicados entre otras tareas a la detección de patologías [6,7,8,9]. El método tradicional de entrenamiento de los HMM es mediante el algoritmo de *Esperanza y Maximización-EM* que es un método de estimación de parámetros que cae dentro del campo general de estimación de máxima verosimilitud (*maximum likelihood - ML*), el cual tiene sus raíces en la teoría clásica de decisión Bayesiana. Si se evalúan las suposiciones fundamentales y limitaciones de ésta aproximación, se puede encontrar que existen diferencias entre el problema de estimar una distribución óptima y el problema de diseñar un sistema de reconocimiento óptimo [10]. Considerar entonces, un sistema de reconocimiento de patrones que emplee en la etapa de extracción de características un método tradicional (con los problemas mencionados antes) y adicionalmente utilice en la etapa de clasificación un HMM entrenado a partir del criterio ML, es un sistema que no está diseñado para obtener óptimo desempeño de reconocimiento. Una forma directa de superar el problema de la inconsistencia entre la etapa de EC y la etapa de clasificación, es conducirlas conjuntamente utilizando un criterio común de optimización [1]. Dado que en muchos problemas de reconocimiento de patrones y específicamente en el procesamiento de bioseñales, se han obtenido mejores resultados a partir de la inclusión de los HMM como método de clasificación, y teniendo en cuenta los problemas comentados antes, se hace necesario desarrollar una metodología de entrenamiento que permite reducir el espacio de los datos de entrada y ajustar los parámetros de los modelo ocultos de Markov de manera conjunta, utilizando un criterio común de reducción de error.

Objetivos

Objetivo General

Desarrollar una metodología de entrenamiento de modelos ocultos de Markov empleando el criterio de mínimo error de clasificación y el uso de una medida de distancia entre modelos, que pueda ser empleada en reducción de espacios de entrenamiento de características dinámicas, para mejorar el desempeño de los sistemas de identificación de patologías en bioseñales.

Objetivos Específicos

- Evaluar diferentes medidas de distancia entre modelos de Markov, a partir de los criterios de valor cuantitativo de separación y de consistencia, para ser utilizada en el algoritmo de entrenamiento de los modelos.
- Desarrollar un algoritmo de estimación de parámetros para modelos de Markov a partir del criterio de mínimo error de clasificación para reducción de dimensión, que incluya el empleo de una medida de distancia para disminuir la similitud entre modelos de clases diferentes.
- Validar la metodología de entrenamiento de los modelos en la reducción de espacios de características en el reconocimiento de patologías en bioseñales, empleando una metodología robusta que mejore la calidad y claridad de los resultados de validación, utilizando matrices de contingencia y curvas de desempeño.

Parte II

Marco teórico

Capítulo 1

Métodos de Entrenamiento de Modelos Ocultos de Markov

1.1. Definición de modelos ocultos de Markov

Una cadena de Markov es un proceso aleatorio $\theta(t)$ que puede tomar una cantidad finita K de valores discretos dentro del conjunto $\{\vartheta_1, \ldots, \vartheta_K\}$, tal que en los momentos determinados del tiempo $(t_0 < t_1 < t_2 < \ldots)$ los valores del proceso aleatorio cambien (con probabilidades de cambio conocidas), esto es, se efectúan los cambios en forma de secuencia aleatoria $\theta_0 \rightarrow \theta_1 \rightarrow \theta_2, \ldots$, siendo $\theta_n = \theta(t_n)$ el valor de la secuencia después del intervalo n de tiempo. Las cadenas de Markov asumen una cantidad finita de valores discretos o estados para la representación de una señal aleatoria. En particular, cada estado de manera directa se asocia a un evento físico observable. Sin embargo, en la práctica, se tienen aplicaciones con señales que no presentan de forma evidente los eventos sobre los cuales se construye el modelo. En este sentido, se debe construir un modelo probabilístico sobre los estados no observables u *ocultos*. Como resultado las cadenas construidas por este principio, corresponden a un proceso estocástico doble incrustado; la función probabilística de los estados ocultos y el mismo modelo de aleatoriedad de Markov impuesto sobre la señal.
Los modelos ocultos de Markov, se pueden caracterizar mediante los siguientes parámetros [11]:

(a). Los símbolos de observación corresponden a la salida física del sistema en análisis y conforman la secuencia aleatoria $\boldsymbol{\varphi} = \{\varphi_1, \ldots, \varphi_{n_\varphi}\}$ en los momentos definidos de tiempo $1, 2, \ldots, n_\varphi$, donde n_φ es la longitud de la secuencia de observación.

(b). El número de los estados ocultos del modelo, $\boldsymbol{\vartheta} = \{\vartheta_k : k = 1, \ldots, n_\vartheta\} \in \mathfrak{V}$, que siendo no observables, pueden ser relacionados con algún sentido físico del proceso. A partir de los estados ocultos se puede establecer una secuencia de estados $\boldsymbol{\theta} = \{\theta_0, \theta_1, \ldots, \theta_{n_\theta}\}$, de todos los posibles estados, en los momentos definidos de tiempo $= 1, 2, \ldots, n_\theta$, donde n_θ es la longitud de la secuencia de estados en análisis.

(c). La matriz probabilidad de transición de estados, $\boldsymbol{\Pi} = \{\pi_{mn} : m, n = 1, \ldots, n_\vartheta\}$, en

la cual cada elemento se determina como,

$$\begin{aligned} \pi_{mn}(k) &= P\left(\theta_{k+1} = \vartheta_n | \theta_k = \vartheta_m\right), \\ \pi_{mn} &\geq 0, \\ \sum_{n=1}^{n_\vartheta} \pi_{mn} &= 1 \end{aligned} \tag{1.1}$$

(d). El conjunto completo de parámetros que representar la distribución de las observaciones por cada estado del modelo $\mathbf{B} = \{b_j(\cdot)\}$.
Existen dos formas de distribuciones de salida que pueden ser consideradas. La primera es una suposición de observación discreta donde se asume que una observación es una de n_υ posibles símbolos de observación $\boldsymbol{\upsilon} = \{\upsilon_k : k = 1, \ldots, n_\upsilon\} \in \mathfrak{U}$. En este caso $b_j(\varphi_n = \upsilon_k) = b_j(\upsilon_k) = p(\upsilon_k | \theta_n = \vartheta_j)$. La segunda forma de distribución de probabilidad, es considerar un mezcla de M funciones de distribución para cada estado. Convencionalmente las funciones utilizadas son Gaussianas multivariadas, debido a sus propiedades y a que todo el aparato matemático está descrito para éstas. En este caso $b_j(\varphi_n) = \sum_{r=1}^{M} c_{jr} \mathcal{N}(\varphi_n | \mu_{jr}, \Sigma_{jr})$, donde μ_{jr} es el vector de medias de la componente normal r en el estado j, Σ_{jr} es la matriz de covarianza de la componente normal r en el estado j y c_{jr} es el peso que pondera la componente Gaussiana r del estado j.
Los modelos que asumen la primera forma de distribución, son llamados modelos ocultos de Markov *discretos*, mientras que los modelos que asumen la segunda forma de distribución de salida son llamados modelos ocultos de Markov *continuos*.

(e). El vector probabilidad de estado inicial $\mathbf{p}_{\theta_1}$ con elementos $\{P_{\theta_1}(i)\}$, donde

$$p_{\theta_1}(i) = P(\theta_1 = \vartheta_i), \ 1 \leqslant i \leqslant n_\vartheta$$

Los valores de aleatoriedad $\boldsymbol{\Pi}$, $\mathbf{B}$ y $\mathbf{p}_{\theta_1}$, notados en conjunto como

$$\lambda = \{\boldsymbol{\Pi}, \mathbf{B}, \mathbf{p}_{\theta_1}\}$$

conforman los parámetros de un modelo oculto de Markov, el cual se puede emplear para generar la estimación de la secuencia de observación, $\boldsymbol{\varphi} \in \{\varphi_1, \varphi_2, \ldots, \varphi_{n_\varphi}\}$, con longitud $n_\varphi = n_\vartheta$ para los momentos definidos de tiempo $n = 1, 2, \ldots, n_\varphi$.
El desarrollo de los modelos ocultos de Markov está relacionado con las siguientes tres tareas estadísticas [11]:

1. Dada una secuencia de observación $\boldsymbol{\varphi} = \{\varphi_1, \varphi_2, \ldots, \varphi_{n_\varphi}\}$ con longitud n_φ y el modelo $\lambda = \{\boldsymbol{\Pi}, \mathbf{B}, \mathbf{p}_{\theta_1}\}$, cómo calcular de manera eficiente la probabilidad $P(\boldsymbol{\varphi}|\lambda)$ de la secuencia de observación.

2. Dada una secuencia de observación $\boldsymbol{\varphi} = \{\varphi_1, \varphi_2, \ldots, \varphi_{n_\varphi}\}$ con longitud n_φ y el modelo conocido λ, cómo escoger de forma óptima la correspondiente secuencia de estados $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_{n_\varphi}\}$ para un criterio de medida fijado a priori.

3. El ajuste de los parámetros del modelo λ que brinden el máximo valor de $P(\boldsymbol{\varphi}|\lambda)$.

Debido al objeto de estudio de este trabajo, nos centraremos en la solución al tercer problema de los citados anteriormente.
Los métodos de entrenamiento de los modelos de Markov se pueden clasificar en dos grupos: *(i) algoritmos de optimización o búsqueda ascendente* y *(ii) algoritmos de búsqueda global* [12]. Los algoritmos de búsqueda ascendente dependen enormemente de la manera en la que se inicialice el modelo, de tal forma que, en la práctica y si los parámetros iniciales no han sido los óptimos, la búsqueda puede conducir a un modelo sub-óptimo. Para evitar este problema se proponen una serie de técnicas aunque estas impliquen una mayor carga computacional. Por otra parte, los algoritmos de búsqueda global no dependen en exceso de la inicialización del modelo, precisamente por su capacidad global para encontrar el óptimo pero presentan problemas de costo computacional.

1.2. Máxima esperanza - EM

El algoritmo de máxima esperanza (*Expectation Maximization - EM*) es un método de estimación de parámetros que cae dentro del campo general de estimación de máxima verosimilitud (*maximum likelihood - ML*) y es aplicado en casos donde parte de los datos pueden ser considerados incompletos u ocultos. Este es esencialmente un algoritmo de optimización iterativo que, bajo ciertas restricciones puede hacer converger los valores de los parámetros a un máximo local de la función de verosimilitud [13]. Existen dos principales aplicaciones del algoritmo EM [14]. La primera ocurre cuando los datos tienen valores perdidos, debido a problemas o limitaciones de los procesos de observación. La segunda ocurre cuando optimizar la función de verosimilitud es analíticamente intratable, pero además la función de verosimilitud puede ser simplificada asumiendo la existencia y los valores de los parámetros perdidos (u ocultos). La última aplicación es más común en problemas computacionales de reconocimiento de patrones.
Existen diversas modificaciones del algoritmo EM, éstas se realizan con base en las restricciones para completar los datos incompletos [15].
El algoritmo EM asume el siguiente problema: Si se tienen dos espacios de muestras $\mathcal{X}$ y $\mathcal{Y}$, tal que se puede realizar un mapeo $\mathbf{X} = f(\mathbf{Y})$ de una observación $\mathbf{Y}$ del espacio $\mathcal{Y}$ a una observación $\mathbf{X}$ en el espacio $\mathcal{X}$. Se define:

$$\mathcal{Y}(\mathbf{X}) = \{\mathbf{Y} : f(\mathbf{Y}) = \mathbf{X}\} \tag{1.2}$$

$\mathbf{Y}$ son los datos *completos*, y $\mathbf{X}$ son los datos *observados*. Si la distribución $f(\mathbf{Y}|\mathbf{\Theta})$ esta bien definida, entonces la probabilidad de $\mathbf{X}$ dado $\mathbf{\Theta}$ es

$$\mathcal{G}(\mathbf{X}|\mathbf{\Theta}) = \int_{\mathcal{Y}(\mathbf{X})} f(\mathbf{X}|\mathbf{\Theta})\, d\mathbf{Y} \tag{1.3}$$

El algoritmo EM esta dirigido a encontrar un valor de $\mathbf{\Theta}$ que maximice $\mathcal{G}(\mathbf{X}|\mathbf{\Theta})$ dado un $\mathbf{X}$ observado, pero para hacer esto, usa esencialmente la familia $f(\mathbf{X}|\mathbf{\Theta})$ asociada. El conjunto de parámetros $\mathbf{\Theta}$ permiten definir la verosimilitud de los datos como $P(\mathbf{Y}|\mathbf{\Theta})$. Puede definirse también el *log* de la verosimilitud $\mathcal{L}(\mathbf{Y}|\mathbf{\Theta}) = \log P(\mathbf{Y}|\mathbf{\Theta})$.

Si Ω es el espacio de los parámetros, la estimación de máxima verosimilitud, consiste en ajustar el estimado $\boldsymbol{\Theta}_{ML}$ tal que

$$\boldsymbol{\Theta}_{ML} = \arg \max_{\boldsymbol{\Theta} \in \Omega} \mathcal{L}\left(\mathbf{Y}|\boldsymbol{\Theta}\right) \tag{1.4}$$

Para el caso de $\mathcal{G}$, se intenta encontrar un conjunto $\boldsymbol{\Theta}$ que maximice

$$\mathcal{L}\left(\boldsymbol{\Theta}\right) = \log \mathcal{G}\left(\mathbf{X}|\boldsymbol{\Theta}\right) \tag{1.5}$$

El algoritmo EM primero encuentra el valor esperado del *log*-verosimilitud de los datos completos $\mathcal{L}\left(\mathbf{Y}|\boldsymbol{\Theta}\right)$ con respecto a los datos desconocidos, por medio de los datos observados $\mathbf{X}$ y de los actuales parámetros estimados. Se define [16]:

$$Q\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i-1)}\right) = E\left[\log p\left(\mathbf{Y}|\boldsymbol{\Theta}\right)|\mathbf{X}, \boldsymbol{\Theta}^{(i-1)}\right] \tag{1.6}$$

donde $\boldsymbol{\Theta}^{(i-1)}$ son los actuales parámetros estimados que se usan para evaluar la esperanza y $\boldsymbol{\Theta}$ son los nuevos parámetros que se optimizan para incrementar Q. La clave para entender la expresión (1.6), esta en que $\mathbf{X}$ y $\boldsymbol{\Theta}^{(i-1)}$ son constantes y $\boldsymbol{\Theta}$ es una variable normal que se desea ajustar. La evaluación de esta esperanza es llamada el paso E del algoritmo. Note el significado de los dos argumentos de la función $Q\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}'\right)$. El primer argumento Θ corresponde a los parámetros que se intentan optimizar intentando maximizar la verosimilitud. El segundo argumento $\boldsymbol{\Theta}'$ corresponde a los parámetros utilizados para evaluar la esperanza.
El segundo paso (paso M) del algoritmo EM busca maximizar la esperanza calculada en el primer paso. Esto es encontrar:

$$\boldsymbol{\Theta}^{(\mathbf{i})} = \arg \max_{\boldsymbol{\Theta}} \mathbf{Q}\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(\mathbf{i-1})}\right) \tag{1.7}$$

Estos dos pasos son repetidos cuanto sea necesario. El objetivo es que el algoritmo haga converger los parámetros $\boldsymbol{\Theta}^{(i)} = \boldsymbol{\Theta}_{ML}$. En cada iteración del algoritmo esta garantizado el incremento del *log*-verosimilitud; por lo tanto el algoritmo converge a un máximo local de la función de verosimilitud [14].
Una modificación del paso M, es que en lugar de maximizar $Q\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(\mathbf{i-1})}\right)$, se encuentra algún $\boldsymbol{\Theta}^{(i)}$ tal que $Q\left(\boldsymbol{\Theta}^{(i)}, \boldsymbol{\Theta}^{(i-1)}\right) > Q\left(\boldsymbol{\Theta}^{(i-1)}, \boldsymbol{\Theta}^{(i-2)}\right)$. Este algoritmo es llamado el EM Generalizado (*Generalized EM - GEM*) y también garantiza la convergencia [17]. La figura 1.1 muestra gráficamente el proceso iterativo del algoritmo EM.

1.2.1. Estimación de parámetros para máxima verosimilitud de modelos ocultos de Markov

Para el caso de cadenas ocultas de Markov la función de verosimilitud de los datos incompletos esta dada por $P\left(\boldsymbol{\varphi}|\lambda\right)$ y la función de verosimilitud de los datos completos esta dada por $P\left(\boldsymbol{\varphi}, \boldsymbol{\theta}|\lambda\right)$. La función Q es por consiguiente:

$$Q\left(\lambda, \lambda'\right) = \sum_{\boldsymbol{\theta} \in \mathcal{Q}} \log P\left(\boldsymbol{\varphi}, \boldsymbol{\theta}|\lambda\right) P\left(\boldsymbol{\varphi}, \boldsymbol{\theta}|\lambda'\right) \tag{1.8}$$

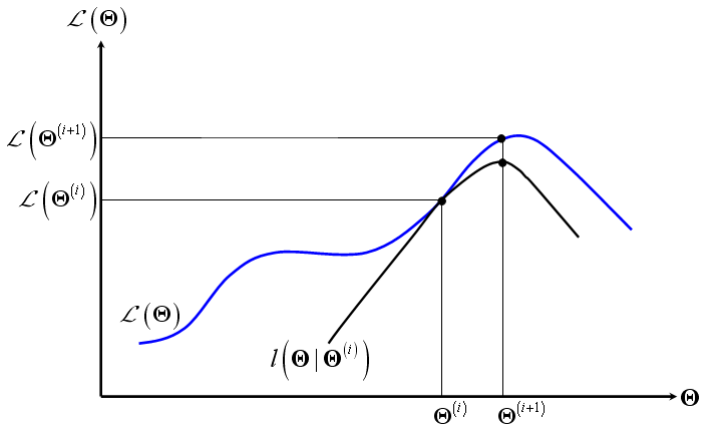


Figura 1.1: Interpretación gráfica de una iteración del algoritmo EM: La función $l\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)}\right)$ está limitada por la función *log*-verosimilitud. Las funciones son iguales en $\boldsymbol{\Theta} = \boldsymbol{\Theta}^{(i)}$. El algoritmo EM escoge $\boldsymbol{\Theta}^{(i+1)}$ como el valor de $\boldsymbol{\Theta}$ para el cual $l\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)}\right)$ es un máximo. Dado que $\mathcal{L}(\boldsymbol{\Theta}) \geq l\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i)}\right)$ y l es incrementada, se asegura que la función de verosimilitud se incremente en cada paso

donde λ' es la estimación de parámetros previa y $\mathcal{Q}$ es el espacio de todas las secuencias de estado de longitud n_θ.
Dada una secuencia de estados particular $\boldsymbol{\theta}^k$, $P\left(\varphi, \boldsymbol{\theta}^k|\lambda\right)$ se puede expresar como:

$$P\left(\boldsymbol{\varphi}, \boldsymbol{\theta}^k|\lambda\right) = \boldsymbol{p}_{\theta_0^k} \prod_{n=1}^{n_\theta} \pi_{\theta_{n-1}^k \theta_n^k} b_{\theta_n^k}\left(\varphi_n\right) \tag{1.9}$$

La función Q puede entonces expresarse como [14]:

$$\begin{aligned} Q\left(\lambda, \lambda'\right) &= \sum_{\theta \in \mathcal{Q}} \log \boldsymbol{p}_{\theta_0^k} P\left(\boldsymbol{\varphi}, \boldsymbol{\theta}^k|\lambda'\right) + \sum_{\theta \in \mathcal{Q}} \left(\sum_{n=1}^{n_\theta} \log \pi_{\theta_{n-1}^k \theta_n^k} \right) P\left(\boldsymbol{\varphi}, \boldsymbol{\theta}^k|\lambda'\right) \\ &+ \sum_{\theta \in \mathcal{Q}} \left(\sum_{n+1}^{n_\theta} \log b_{\theta_n^k}\left(\varphi_n\right) \right) P\left(\boldsymbol{\varphi}, \boldsymbol{\theta}^k|\lambda'\right) \end{aligned} \tag{1.10}$$

Por consiguiente puede ser optimizado cada término individualmente. El primer término en la ecuación (1.10) se puede expresar como:

$$\sum_{\theta \in \mathcal{Q}} \log \boldsymbol{p}_{\theta_0^k} P\left(\boldsymbol{\varphi}, \boldsymbol{\theta}^k|\lambda'\right) = \sum_{i=1}^{n_\vartheta} \log \boldsymbol{p}_{\theta_0^k}(i) P\left(\boldsymbol{\varphi}, \theta_0^k = i|\lambda'\right) \tag{1.11}$$

el lado derecho es sólo la expresión marginal para el tiempo $n = 0$. Adicionando el multiplicador de lagrange γ, usando la restricción $\sum_i \boldsymbol{p}_{\theta_0^k}(i) = 1$, derivando e igualando a cero, se tiene:

$$\frac{\partial}{\partial \boldsymbol{p}_{\theta_0^k}(i)} \left(\sum_{i=1}^{n_\vartheta} \log \boldsymbol{p}_{\theta_0^k}(i) P\left(\boldsymbol{\varphi}, \theta_0^k = i|\lambda'\right) + \gamma \left(\sum_{i=1}^{n_\vartheta} \boldsymbol{p}_{\theta_0^k}(i) - 1 \right) \right) = 0 \tag{1.12}$$

Resolviendo para $\boldsymbol{p}_{\theta_0^k}$, se tiene [14]:

$$\boldsymbol{p}_{\theta_0^k}(i) = \frac{P\left(\boldsymbol{\varphi}, \theta_0^k = i|\lambda'\right)}{P\left(\boldsymbol{\varphi}|\lambda'\right)} \tag{1.13}$$

El segundo término en la ecuación (1.10) se puede escribir como:

$$\sum_{\theta\in\mathcal{Q}}\left(\sum_{n=1}^{n_\theta}\log\pi_{\theta_{n-1}^k\theta_n^k}\right)P\left(\boldsymbol{\varphi}, \theta^k|\lambda'\right) = \sum_{i=1}^{n_\vartheta}\sum_{j=1}^{n_\vartheta}\sum_{n=1}^{n_\theta}\log\pi_{ij}P\left(\boldsymbol{\varphi}, \theta_{n-1}^k = i, \theta_n^k = j|\lambda'\right) \tag{1.14}$$

en esta expresión se evalúa en cada tiempo n todas las posibles transiciones de i a j, y se ponderan por su correspondiente probabilidad. De manera similar al procedimiento realizado para el primer término de (1.10), se puede usar un multiplicador de lagrange y empleando la restricción (1.1), se tiene:

$$\pi_{ij} = \frac{\sum_{n=1}^{n_\theta}P\left(\boldsymbol{\varphi}, \theta_{n-1}^k = i, \theta_n^k = j|\lambda'\right)}{\sum_{t=1}^{n_\theta}P\left(\boldsymbol{\varphi}, \theta_{n-1}^k = i|\lambda'\right)} \tag{1.15}$$

El tercer término en la ecuación (1.10), se puede escribir como:

$$\sum_{\theta\in\mathcal{Q}}\left(\sum_{n+1}^{n_\theta}\log b_{\theta_n^k}(\varphi_n)\right)P\left(\boldsymbol{\varphi}, \boldsymbol{\theta}^k|\lambda'\right) = \sum_{i=1}^{n_\vartheta}\sum_{n=1}^{n_\theta}\log b_i(\varphi_n)P\left(\boldsymbol{\varphi}, \theta_n^k = i|\lambda'\right) \tag{1.16}$$

Interpretando la ecuación, se observa que para cada tiempo n, se evalúan las emisiones para todos los estados y se pondera cada una de las posibles emisiones por su correspondiente probabilidad. Para distribuciones discretas, se puede una vez más, utilizar el multiplicador de lagrange con la restricción $\sum_{j=1}^{n_\upsilon}b_i(\upsilon_j) = 1$. Únicamente las observaciones que son iguales a υ_k contribuyen al k^{th} valor de probabilidad, por lo tanto:

$$b_i(k) = \frac{\sum_{n=1}^{n_\theta}P\left(\varphi, \theta_{n-1}^k = i|\lambda'\right)\delta_{\upsilon_j,\upsilon_k}}{\sum_{t=1}^{n_\theta}P\left(\varphi, \theta_{n-1}^k = i|\lambda'\right)} \tag{1.17}$$

donde δ es la función delta de Kronecker.
Para distribuciones continuas (mezclas de gaussianas), la forma de la función Q es ligeramente diferente, las variables ocultas deben incluir no solo la secuencia de estados oculta, si no también una variable que indica la componente de la mezcla para cada estado en cada tiempo. Por consiguiente, se puede escribir Q como:

$$Q(\lambda, \lambda') = \sum_{\theta\in\mathcal{Q}}\sum_{m\in\mathcal{M}}\log P\left(\boldsymbol{\varphi}, \boldsymbol{\theta}^k, m|\lambda\right)P\left(\boldsymbol{\varphi}, \boldsymbol{\theta}^k, m|\lambda'\right) \tag{1.18}$$

donde $\mathcal{M}$ es el conjunto de todas las componentes y $m = \left\{m_{\theta_1^k 1}, m_{\theta_2^k 2}, ..., m_{\theta_{n_\theta}^k n_\theta}\right\}$ es el vector que indica la componente de mezcla para cada estado y cada tiempo. Si se expande (1.18) de igual manera como la ecuación (1.10), el primero y segundo término no cambian

debido a que los parámetros son independientes de m. El tercer término en la ecuación (1.18) sería:

$$\sum_{\theta\in\mathcal{Q}}\sum_{m\in\mathcal{M}}\left(\sum_{n=1}^{n_\theta}\log b_{\theta_n^k}\left(\varphi_n, m_{\theta_n^k n}\right)\right)P\left(\boldsymbol{\varphi},\boldsymbol{\theta}^k, m|\lambda'\right)=\sum_{i=1}^{n_\vartheta}\sum_{r=1}^{M}\sum_{n=1}^{n_\theta}\log\left(c_{ir}b_{ir}\left(\varphi_n\right)\right)P\left(\varphi,\theta_n^k=i,m_{\theta_n^k n}=r|\lambda'\right) \tag{1.19}$$

Para optimizar esta expresión, se deben maximizar cada término de las variables "*ocultas*"por separado. Se puede obtener [14]:

$$c_{ir}=\frac{\sum_{n=1}^{n_\theta}P\left(\theta_n^k=i,m_{\theta_n^k n}=r|\boldsymbol{\varphi},\lambda'\right)}{\sum_{n=1}^{n_\theta}\sum_{r=1}^{M}P\left(\theta_n^k=i,m_{\theta_n^k n}=r|\boldsymbol{\varphi},\lambda'\right)}, \tag{1.20}$$

$$\mu_{ir}=\frac{\sum_{n=1}^{n_\theta}\varphi_n P\left(\theta_n^k=i,m_{\theta_n^k n}=r|\boldsymbol{\varphi},\lambda'\right)}{\sum_{n=1}^{n_\theta}P\left(\theta_n^k=i,m_{\theta_n^k n}=r|\boldsymbol{\varphi},\lambda'\right)}, \tag{1.21}$$

y

$$\Sigma_{ir}=\frac{\sum_{n=1}^{n_\theta}\left(\varphi_n-\mu_{ir}\right)\left(\varphi_n-\mu_{ir}\right)^T P\left(\theta_n^k=i,m_{\theta_n^k n}=r|\boldsymbol{\varphi},\lambda'\right)}{\sum_{n=1}^{n_\theta}P\left(\theta_n^k=i,m_{\theta_n^k n}=r|\boldsymbol{\varphi},\lambda'\right)}$$

El algoritmo dinámico que realiza el ajuste de los parámetros de manera computacionalmente óptima, es el algoritmo de Baum-Welch [11, 14].

1.3. Entrenamiento discriminativo

El método de entrenamiento más popular para sistemas basados en HMM es la estimación de máxima verosimilitud. Este método está basado en la teoría clásica de decisión de Bayes, en la cual se asume que un clasificador óptimo es aquel que implementa una regla de clasificación de la forma [10]:

$$C\left(\varphi\right)=c_i \quad \text{si} \quad P\left(c_i|\varphi\right)=\max_j P\left(c_j|\varphi\right) \tag{1.22}$$

donde φ es una observación arbitraria y $C\left(\varphi\right)$ representa la decisión de clasificación. En otras palabras, una observación φ es etiquetada a la clase i, si la probabilidad *a posteriori* $P\left(c_i|\varphi\right)$ es mayor que para cualquier otra clase c_j. De esta manera se asume que el mínimo error de clasificación es alcanzado cuando se utiliza la regla (1.22). Este criterio de decisión es llamado el *maximum a posteriori (MAP)*. El mínimo error de clasificación alcanzado por la decisión MAP es llamado el *riesgo de Bayes*. Para una óptima clasificación por medio de la regla *MAP* se requiere el conocimiento exacto de las probabilidades condicionales $P\left(c_j|\varphi\right),\ \ j=1,...,N$, donde N es el número de clases. Sin embargo, en la práctica estas probabilidades no son dadas y deben ser estimadas a partir del conjunto de datos de entrenamiento. La teoría de decisión de Bayes transforma entonces el problema de diseño de un clasificador en un problema de estimación de distribución [10]. En muchos problemas reales estimar la distribución de los datos es una tarea difícil, debido en primer lugar, a que en la mayoría de los casos se hace una estimación paramétrica de la distribución de los datos y está requiere seleccionar la forma de la distribución; esta selección se ve limitada

por la complejidad matemática de funciones de distribución particulares y por tanto es muy probable que se genere inconsistencia con la distribución real de los datos. Este hecho causa que la verdadera regla de decisión MAP pueda en muy pocos casos ser implementada. En segundo lugar, la estimación paramétrica de la forma de la distribución se hace a partir de los datos de entrenamiento, por lo que se hace necesario para obtener una buena estimación, que el tamaño del conjunto de datos de entrenamiento sea suficiente, lo que en muchas tareas de reconocimiento es muy complicado obtener y por lo tanto la calidad de la de la estimación de los parámetros de la distribución no puede ser garantizado [10]. Por estas razones se han generado diversos métodos de entrenamiento de modelos de ocultos de Markov, que buscan corregir los problemas de la aproximación clásica y estiman los parámetros del modelo minimizando directamente una representación adecuada del error de decodificación [18].

1.3.1. Máxima información mutua

Convencionalmente la estimación de máxima verosimilitud intenta incrementar la probabilidad *a posteriori* de los datos de entrenamiento dada la secuencia del modelo correspondiente a los datos. Los modelos de otras clases no participan en la re-estimación de los parámetros, como consecuencia, el criterio ML no relaciona directamente el objetivo de reducción de la tasa de error [19]. Aunque el entrenamiento de modelos ocultos de Markov por el método de Máxima información mutua (*Maximum Mutual Information (MMI)*) busca ajustar una función de distribución, al igual que en el ML, la diferencia radica en el tipo de distribución que se quiere ajustar. La distribución que se quiere ajustar mediante ML es la distribución de cada clase, mientras que en MMI se busca ajustar la distribución posterior de clase condicional (que puede entenderse como una distribución entre clases). Típicamente, las distribuciones de datos son mucho más complejas de describir que las distribuciones posteriores, debido a que las distribuciones de datos deben describir todas las variaciones de los datos dentro de una misma clase, mientras la distribución posterior sólo se ocupa de los límites entre las clases [18].
Considere el conjunto de modelos HMMs [20].

$$\Psi = \{\lambda_i,\ \ 1 \leq i \leq N\} \tag{1.23}$$

la tarea es minimizar la incertidumbre condicional de una clase i dada una secuencia de observaciones $\boldsymbol{\varphi}$ de longitud n_φ perteneciente a la clase i. Esto equivale a minimizar la información condicional:

$$I\left(i|\boldsymbol{\varphi}, \Psi\right) = -\log p\left\{i|\boldsymbol{\varphi}, \Psi\right\} \tag{1.24}$$

En el campo de teoría de la información este problema conduce a la minimización de la entropía condicional, definida como la esperanza $E\left(\cdot\right)$ de la información condicional I,

$$H\left(\iota|\Phi\right) = E\left[I\left(i|\boldsymbol{\varphi}\right)\right] \tag{1.25}$$

donde ι representa todas las clases y Φ representa todas las secuencias de observación. Entonces la información mutua entre las clases y observaciones dada por:

$$H\left(\iota, \Phi\right) = H\left(\iota\right) - H\left(\iota|\Phi\right) \tag{1.26}$$

debe ser maximizada teniendo en cuenta que $H(\iota)$ es constante. Aunque la ecuación (1.24) define el criterio MMI, éste puede ser reescrito usando el teorema de Bayes para obtener una mejor comprensión, como se muestra en la ecuación (1.27) [21]:

$$\begin{aligned} E_{MMI} &= -\log p\{i|\varphi, \Psi\} \\ &= -\log \frac{p\{i,\varphi|\Psi\}}{p\{\varphi|\Psi\}} \\ &= -\log \frac{p\{i,\varphi|\Psi\}}{\sum_{\varsigma} p\{\varsigma,\varphi|\Psi\}} \end{aligned} \tag{1.27}$$

donde ς representa una clase arbitraria. El objetivo entonces es utilizar un método de minimización del criterio E_{MMI} en función del conjunto de parámetros de los HMMs, de tal manera que se maximice la información mutua:

$$H(\iota, \boldsymbol{\varphi}) = H(\iota) \tag{1.28}$$

esto implica que el modelo elimina todas las incertidumbres acerca del etiquetado y de esta manera, maximizar la información mutua puede conducir a un decodificador perfecto [21]. Por varias razones, el entrenamiento de los HMMs mediante MMI es mucho más complejo que con ML. Una razón de ésto es la no existencia de fórmulas de reestimación en forma cerrada similares a aquellas que existen para ML. Un conocido algoritmo utilizado para maximizar el criterio MML es presentado en [22], éste es basado en una generalización de la desigualdad de Baum-Eagon [23]. El método fue inicialmente propuesto para HMM discretos, y generalizado para modelos con densidades Gaussianas en [24] y se conoce como el algoritmo extendido de *Baum - Welch (EBW)*. Aunque el algoritmo EBW ha sido utilizado en diferentes tareas de reconocimiento de voz, presenta algunos defectos que han sido tratados de corregir por otras aproximaciones [23]. El método MMI demuestra ventajas en el desempeño en comparación con la aproximación tradicional ML, sin embargo, no ésta basado en una directa minimización de la función de perdida que está ligada a la tasa de error de clasificación.

1.3.2. Error de clasificación mínimo

El método de entrenamiento empleando el criterio de error de clasificación mínimo (*minimum classification error - MCE*), introducido en [5] y extendido para HMM en [10], busca minimizar la probabilidad de error a través de una representación suavizada de la función de pérdida (loss function) que para el caso en el cual la decisión a tomar es de pertenencia o no una clase específica, se asume zero-uno. Esto se hace a través de la llamada medida del error de clasificación, que representa simplemente una medida de la distancia entre la probabilidad de una decisión correcta y otras decisiones. Existen varias definiciones de esta medida, una de las cuales está dada por:

$$d_i(\boldsymbol{\varphi}) = -g_i(\boldsymbol{\varphi}; \lambda_i) + \log\left[\frac{1}{N-1}\sum_{j,j\neq i} \exp\left[g_j(\boldsymbol{\varphi}; \lambda_j)\,\eta\right]\right]^{1/\eta} \tag{1.29}$$

donde N es el número de clases, η es un número positivo y $g_i(\varphi;\lambda)$ es la función de verosimilitud condicional para la clase i, que para el caso de HMM puede ser utilizada la probabilidad conjunta estado-observación definida por:

$$\begin{aligned} g_i(\boldsymbol{\varphi};\lambda) &= \log\left\{\max_{\boldsymbol{\theta}} g_i(\boldsymbol{\varphi},\boldsymbol{\theta};\lambda)\right\} = \log\left\{g_i\left(\boldsymbol{\varphi},\bar{\boldsymbol{\theta}};\lambda\right)\right\} \\ &= \sum_{n=1}^{n_\varphi}\left[\log \boldsymbol{\Pi}^{(i)}_{\bar{\theta}_{n-1}\bar{\theta}_n} + \log b_j^{(i)}(\varphi_n)\right] + \log \boldsymbol{p}_{\bar{\theta}_1} \end{aligned} \tag{1.30}$$

donde $\bar{\theta}$ corresponde a la secuencia de estados más probable en el modelo para una secuencia de observación dada, y puede ser calculada mediante el algoritmo de Viterbi [11]. En este trabajo se empleo la función de verosimilitud condicional dada por:

$$g_i(\boldsymbol{\varphi};\lambda) = \frac{1}{n_\varphi}\left(\sum_{n=1}^{n_\varphi}\left[\log \boldsymbol{\Pi}^{(i)}_{\bar{\theta}_{n-1}\bar{\theta}_n} + \log b_j^{(i)}(\varphi_n)\right] + \log \boldsymbol{p}_{\bar{\theta}_1}\right) \tag{1.31}$$

la escala $1/n_\varphi$ permite normalizar la función de verosimilitid de la duración de las secuencias [25].

La medida de error de clasificación es una función continua de los parámetros del decodificador e intenta emular la regla de decisión. Para una secuencia de observaciones $\boldsymbol{\varphi}$ que pertenezca a la clase i, $d_i(\boldsymbol{\varphi}) > 0$ implica un error en la clasificación y $d_i(\boldsymbol{\varphi}) \ll 0$ implica una decisión correcta.

Para completar la definición de la función objetivo la medida definida en (1.29) es embebida en una función zero-uno suavizada (que representa la función de pérdida), para la cual cualquier miembro de la familia sigmoidal es un obvio candidato [5]:

$$\ell(d_i(\boldsymbol{\varphi})) = \frac{1}{1+\exp(-\gamma d_i(\boldsymbol{\varphi})+\alpha)} \tag{1.32}$$

donde normalmente α es igual a cero y $\gamma \geq 1$. Claramente se ve que cuando $d_i(\boldsymbol{\varphi}_i)$ es mucho menor que zero, lo que implica una decodificación correcta, no se incurre casi que en ninguna pérdida. Cuando $d_i(\boldsymbol{\varphi}_i)$ es positivo, conduce a una penalización que representa esencialmente una cuenta en el error de decodificación. Finalmente, para cualquier φ desconocida, el desempeño del clasificador/decodificador se mide por:

$$\ell(\boldsymbol{\varphi};\lambda) = \sum_{i=1}^{N} \ell(d_i(\boldsymbol{\varphi}))\, 1(\varphi \in c_i) \tag{1.33}$$

donde $1(\cdot)$ es una función de indicación y es 1 si $(\cdot)$ es verdadero y 0 de otra forma. Esta definición en tres etapas simula la operación de decodificación a la vez que la evaluación del desempeño en un forma funcional suavizada, adecuada para la optimización de los parámetros del clasificador/decodificador. Con base en el criterio de (1.33), se puede elegir minimizar una de dos cantidades para el cálculo de los parámetros del decodificador: la pérdida esperada o la pérdida empírica [10]. Esto se logra mediante técnicas de gradiente descendente; en particular el algoritmo descendente probabilístico generalizado (GPD) [5]. El algoritmo GPD puede ser generalizado de la siguiente forma:

$$\lambda_{n+1} = \lambda_n - \epsilon_n U_n \nabla \ell(\boldsymbol{\varphi}_n,\lambda)\,|_{\lambda=\lambda_n} \tag{1.34}$$

donde U_n es una matriz definida positiva y ϵ es la tasa de aprendizaje [10]. En particular, el algoritmo GPD es un esquema de minimización sin restricciones que necesita modificaciones para resolver un problema de minimización con restricciones, como es el caso del entrenamiento de los HMM. Para esto se utiliza una transformación entre los parámetros minimizados por el algoritmo y los parámetros restringidos de tal manera que se mantenga las restricciones en el espacio original de los parámetros. Una de las ventajas del algoritmo de minimización basado en GPD es que este no hace ninguna suposición explícita sobre las probabilidades desconocidas. Esta propiedad es importante para problemas de reconocimiento y aprendizaje adaptativo [10].
Otro algoritmo desarrollados para resolver la tarea de minimización de la función de pérdida es algoritmo *Quick - prop* [26], que combina una técnica de gradiente descendente y el algoritmo de Newton y usa una aproximación de la matrix Hessiana que no requiere cálculos extra, con el objetivo de aumentar la velocidad de convergencia del algoritmo GPD.
Para utilizar el algoritmo GPD generalizado (1.34), se deben definir las siguientes transformaciones de parámetros que permiten mantener las restricciones probabilísticas de los parámetros de los HMM durante la adaptación:

$$\mathbf{\Pi}_{jk} \to \tilde{\mathbf{\Pi}}_{jk} \quad \text{donde} \quad \mathbf{\Pi}_{jk} = \frac{\exp\left(\tilde{\mathbf{\Pi}}_{jk}\right)}{\sum_{l=1}^{n_\vartheta} \exp\left(\tilde{\mathbf{\Pi}}_{jl}\right)} \tag{1.35}$$

$$\boldsymbol{p}_{\theta_1}(j) \to \tilde{\boldsymbol{p}}_{\theta_1}(j) \quad \text{donde} \quad \boldsymbol{p}_{\theta_1}(j) = \frac{\exp\left(\tilde{\boldsymbol{p}}_{\theta_1}(j)\right)}{\sum_{l=1}^{n_\vartheta} \exp\left(\tilde{\boldsymbol{p}}_{\theta_1}(l)\right)} \tag{1.36}$$

Para la adaptación de parámetros de las componentes Gaussianas del modelo, se asume por simplicidad que la matrix de covarianza $\Sigma_{jr} = [\sigma_{jrp}^2]_{p=1}^{\rho}$ se asume diagonal.

$$c_{jr} \to \tilde{c}_{jr} \quad \text{donde} \quad c_{jr} = \frac{\exp(\tilde{c}_{jr})}{\sum_{l=1}^{M} \exp(\tilde{c}_{jl})} \tag{1.37}$$

$$\mu_{jrp} \to \tilde{\mu}_{jrp} \quad \text{donde} \quad \tilde{\mu}_{jrp} = \frac{\mu_{jrp}}{\sigma_{jrp}} \tag{1.38}$$

$$\sigma_{jrp} \to \tilde{\sigma}_{jrp} \quad \text{donde} \quad \tilde{\sigma}_{jrp} = \log \sigma_{jrp} \tag{1.39}$$

Se puede mostrar que para una secuencia $\boldsymbol{\varphi}_n \in C_i$ del conjunto de entrenamiento, el ajuste discriminativo del parámetro $\tilde{\mathbf{\Pi}}$ partiendo de la definición (1.34), esta dado por:

$$\tilde{\mathbf{\Pi}}_{jk}^{(i)}(n+1) = \tilde{\mathbf{\Pi}}_{jk}^{(i)}(n) - \varepsilon \left.\frac{\partial \ell_i(\varphi_n; \lambda)}{\partial \tilde{\mathbf{\Pi}}_{jk}^{(i)}}\right|_{\lambda=\lambda_n} \tag{1.40}$$

Realizando la derivada parcial de la ecuación (1.40) se obtiene la siguiente formula para la actualización de la matriz de transición de estados (El cálculo completo para el desarrollo de las ecuaciones de reestimación, podrá ser consultado en el apéndice A):

$$\begin{aligned}\tilde{\mathbf{\Pi}}_{jk}^{(i)}(n+1) = \tilde{\mathbf{\Pi}}_{jk}^{(i)}(n) + \epsilon\gamma\ell\left(d_i(\boldsymbol{\varphi}_n)\right)\left(1 - \ell\left(d_i(\boldsymbol{\varphi}_n)\right)\right) \\ \tfrac{1}{n_\varphi} \sum_{t=1}^{n_\varphi} \delta\left(\bar{\theta}_{t-1} - j\right) \delta\left(\bar{\theta}_t - k\right) \left(1 - \mathbf{\Pi}_{jk}^{(i)}(n)\right)\end{aligned} \tag{1.41}$$

donde δ es la función delta de Kronecker. De igual forma, para actualizar el vector de probabilidad de estado inicial, se utiliza la definición (1.34) y se obtiene:

$$\begin{aligned}\tilde{\boldsymbol{p}}_{\theta_1}^{(i)}(j,n+1) = \tilde{\boldsymbol{p}}_{\theta_1}^{(i)}(j,n) + \epsilon\gamma\ell\left(d_i\left(\boldsymbol{\varphi}_n\right)\right)\left(1-\ell\left(d_i\left(\boldsymbol{\varphi}_n\right)\right)\right) \\ \tfrac{1}{n_\varphi}\delta\left(\bar{\theta}_1 - j\right)\left(1-\boldsymbol{p}_{\theta_1}^{(i)}(j,n)\right)\end{aligned} \tag{1.42}$$

Las ecuaciones para actualizar los parámetros de las mezclas gaussianas del modelo, se presentan a continuación:

$$\begin{aligned}\tilde{c}_{jr}^{(i)}(n+1) = \tilde{c}_{jr}^{(i)}(n) + \epsilon\gamma\ell\left(d_i\left(\boldsymbol{\varphi}_n\right)\right)\left(1-\ell\left(d_i\left(\boldsymbol{\varphi}_n\right)\right)\right) \\ \tfrac{1}{n_\varphi}\sum_{t=1}^{n_\varphi}\delta\left(\bar{\theta}_t - j\right)\left(b_j^{(i)}\left(\varphi_t\right)\right)^{-1}\left|\Sigma_{jr}^{(i)}\right|^{-1/2}(2\pi)^{-\rho/2} \\ \exp\left(-\tfrac{1}{2}\sum_{l=1}^{\rho}\left(\tfrac{\varphi_{tl}-\mu_{jrl}^{(i)}(n)}{\sigma_{jrl}^{(i)}(n)}\right)^2\right)c_{jr}^{(i)}\left(1-c_{jr}^{(i)}\right)\end{aligned} \tag{1.43}$$

$$\begin{aligned}\tilde{\mu}_{jrm}^{(i)}(n+1) = \tilde{\mu}_{jrm}^{(i)}(n) + \epsilon\gamma\ell\left(d_i\left(\boldsymbol{\varphi}_n\right)\right)\left(1-\ell\left(d_i\left(\boldsymbol{\varphi}_n\right)\right)\right) \\ \tfrac{1}{n_\varphi}\sum_{t=1}^{n_\varphi}\delta\left(\bar{\theta}_t - j\right)c_{jr}^{(i)}\left(b_j^{(i)}\left(\varphi_t\right)\right)^{-1}\left|\Sigma_{jr}^{(i)}\right|^{-1/2}(2\pi)^{-\rho/2} \\ \exp\left(-\tfrac{1}{2}\sum_{l=1}^{\rho}\left(\tfrac{\varphi_{tl}-\mu_{jrl}^{(i)}(n)}{\sigma_{jrl}^{(i)}(n)}\right)^2\right)\left(\tfrac{\varphi_{tm}}{\sigma_{jrm}^{(i)}(n)} - \tilde{\mu}_{jrm}^{(i)}(n)\right)\end{aligned} \tag{1.44}$$

$$\begin{aligned}\tilde{\sigma}_{jrm}^{(i)}(n+1) = \tilde{\sigma}_{jrm}^{(i)}(n) + \epsilon\gamma\ell\left(d_i\left(\boldsymbol{\varphi}_n\right)\right)\left(1-\ell\left(d_i\left(\boldsymbol{\varphi}_n\right)\right)\right) \\ \tfrac{1}{n_\varphi}\sum_{t=1}^{n_\varphi}\delta\left(\bar{\theta}_t - j\right)c_{jr}^{(i)}\left(b_j^{(i)}\left(\varphi_t\right)\right)^{-1}\left|\Sigma_{jr}^{(i)}\right|^{-1/2}(2\pi)^{-\rho/2} \\ \exp\left(-\tfrac{1}{2}\sum_{l=1}^{\rho}\left(\tfrac{\varphi_{tl}-\mu_{jrl}^{(i)}(n)}{\sigma_{jrl}^{(i)}(n)}\right)^2\right)\left(\left(\tfrac{\varphi_{tm}-\mu_{jrm}^{(i)}(n)}{\sigma_{jrm}^{(i)}(n)}\right)^2 - 1\right)\end{aligned} \tag{1.45}$$

De igual forma, el desarrollo completo para la determinación de las ecuaciones de reestimación de los parámetros, se presenta en el apéndice A.

Capítulo 2

Comparación de modelos ocultos de Markov

Uno de los principales problemas que debe ser abordado cuando se emplean modelos ocultos de Markov, es la estrategia de clasificación de una nueva secuencia de entrada. La manera convencional de clasificación, es encontrar la máxima probabilidad a posteriori de que un modelo particular λ, genere la secuencia de observación φ que se desea etiquetar. De esta manera, si se tienen dos modelos λ_1 y λ_2 (cada uno perteneciente a una clase particular), la secuencia nueva será asignada a la clase del modelo que generó la mayor probabilidad a posteriori. Así

$$C(\varphi) = \begin{cases} 1, & \text{si} \quad p(\varphi|\lambda_1) > p(\varphi|\lambda_2) \\ 2, & \text{si} \quad p(\varphi|\lambda_2) > p(\varphi|\lambda_1) \end{cases} \tag{2.1}$$

donde $C(\cdot)$ es la clase asignada a la secuencia φ. En otras palabras, la probabilidad de que un HMM genere una secuencia dada, indica qué tan probable es que ésta sea miembro de la familia de secuencias modeladas por el HMM, y la secuencia de estados más probable, asociada a la secuencia de observaciones, corresponde al "alineamiento" de la secuencia en relación con la familia de secuencias modeladas [27].

La forma de clasificación descrita en (2.1), aunque ha sido empleada en diversas aplicaciones que involucran el empleo de HMM's [11, 21, 28, 29, 30], no da una medida de la semejanza (o diferencia) entre los mismos. El objetivo de establecer un métrica que refleje de manera cuantitativa la diferencia entre dos HMM's, parte de la necesidad de comparar dos trayectorias dinámicas (secuencias estocásticas), que típicamente son modelados a partir de HMM's [31]. Por consiguiente, obtener una métrica que permita establecer la similitud entre modelos, permite implícitamente comparar la similitud entre dos trayectorias (secuencias).

Una definición alternativa presentada en [32], define el problema de comparación, como el de asignar una distancia a un par de sistemas llamados secuencias

$$\boldsymbol{\varphi}_1 = \langle \varphi_{1,1}, \varphi_{1,2}, ..., \varphi_{1,n_{\varphi 1}} \rangle \qquad \boldsymbol{\varphi}_2 = \langle \varphi_{2,1}, \varphi_{2,2}, ..., \varphi_{2,n_{\varphi 2}} \rangle \tag{2.2}$$

emitidas por dos procesos, mientras procesaban la misma entrada. Cada $\varphi_{i,j}$ denota la salida del *j-ésimo* sistema ante la *i-ésima* entrada. La distancia podría indicar si estás

secuencias reflejan actividades similares.
Dados dos procesos aleatorios de Markov de primer orden λ_1 y λ_2 se debe hallar una medida de similitud o distancia entre los mismos, $d(\lambda_1, \lambda_2)$, con el fin de medir su equivalencia estadística. En la práctica, se han propuesto diversas medidas de distancia. La primer medida propuesta corresponde a una forma generalizada de la distancia euclídea entre las matrices de probabilidad estado-observación, que se define como:

$$d(\lambda_1, \lambda_2) = \sqrt{\frac{1}{n_\vartheta} \sum_{j=1}^{n_\vartheta} \sum_{k=1}^{n_\upsilon} \| b_j^{(1)}(\upsilon_k) - b_j^{(2)}(\upsilon_k) \|^2}, \tag{2.3}$$

Un caso más general de esta medida se realiza encontrando el estado del segundo proceso que minimiza la diferencia entre las probabilidades del los modelos,

$$d(\lambda_1, \lambda_2) = \left\{ \frac{1}{n_\vartheta n_\upsilon} \sum_{j=1}^{n_\vartheta} \sum_{k=1}^{n_\upsilon} \left(b_j^{(1)}(\upsilon_k) - b_{s(j)}^{(2)}(\upsilon_k) \right)^2 \right\}^{1/2} \tag{2.4}$$

donde $s(j)$ es la permutación de los estados que minimiza (2.4). Las medidas (2.3) y (2.4) son inadecuadas, dado que no toman en cuenta la estructura temporal representada en la cadena de Markov, por lo que podría darse el caso en el cual es posible encontrar un par de modelos ocultos de Markov, con una distancia entre sí que tienda a cero, pero con medidas respectivas de probabilidad, P_λ y $P_{\lambda'}$, completamente diferentes [33]. Además del problema antes descrito para la medida basada en la distancia euclídea, existe una razón más para descartar su utilización en este trabajo, y es debido a que esta media está orientada a comparar HMM *discretos*, por lo que no puede ser empleada en la medición de modelos *continuos*, que son precisamente, los modelos empleados por el algoritmo MCE, utilizado en este trabajo.
Una medida alterna presentada en [34], permite comparar dos modelos HMM, utilizando toda la información contenida en el modelos. La media fue desarrollada con base en la distancia de *Kullback-Leibler* entre dos funciones de densidad de probabilidad $p_1(\varphi)$ y $p_2(\varphi)$. Para entender mejor la definición de la medida de distancia entre HMM descrita en [34], se revisará primero la distancia de *Kullback-Leibler*.

2.1. Distancia de *Kullback-Leibler*

La distancia de *Kullback-Leibler* puede ser usada para juzgar qué tan cercanas son dos funciones de densidad de probabilidad $p_1(\varphi)$ y $p_2(\varphi)$. La medida que determina qué tan cercana es la función de densidad $p_2(\varphi)$ a $p_1(\varphi)$ con respecto a $p_1(\varphi)$ es [35]:

$$I(p_1, p_2) = \int_{-\infty}^{\infty} p_1(\varphi) \log \left(\frac{p_1(\varphi)}{p_2(\varphi)} \right) d\varphi \tag{2.5}$$

En el caso en que $p_1(\varphi)$ y $p_2(\varphi)$ son funciones de probabilidad de masa, entonces

$$I(p_1, p_2) = \sum_{\forall \varphi} p_1(\varphi) \log \left(\frac{p_1(\varphi)}{p_2(\varphi)} \right) \tag{2.6}$$

Se debe tener en cuenta que la distancia *Kullback-Leibler* no es simétrica $I(p_1, p_2) \neq I(p_2, p_1)$. Si el objetivo es simplemente comparar p_1 y p_2 se puede definir una medida de distancia simétrica como [35]:

$$I_s = \frac{1}{2}\left[I(p_1, p_2) + I(p_2, p_1)\right] \tag{2.7}$$

El típico problema de aproximación, es que dada la función de densidad $p(\varphi)$, cómo se puede aproximar está con otra función de densidad $\widehat{p}(\varphi)$ (donde $\widehat{p}(\varphi)$ puede ser una función parametrizada). Entonces para obtener $\widehat{p}(\varphi)$, se puede considerar el problema

$$\widehat{p}^{\,*} = \arg\min_{\widehat{p}} \left[I\left(p, \widehat{p}\right)\right] \tag{2.8}$$

Usando $I\left(p, \widehat{p}\right) = \int_\varphi p(\varphi) \log p(\varphi) - \int_\varphi p(\varphi) \log \widehat{p}(\varphi)$ la minimización de la ecuación (2.7) es equivalente a:

$$\widehat{p}^{\,*} = \arg\max_{\widehat{p}} \int_\varphi p(\varphi) \log \widehat{p}(\varphi) \tag{2.9}$$

Por consiguiente, $I\left(p, \widehat{p}\right)$ puede también ser interpretada como:

$$I\left(p, \widehat{p}\right) = E_{p(\varphi)}\left[\log p(\varphi)\right] - E_{p(\varphi)}\left[\log \widehat{p}(\varphi)\right] \tag{2.10}$$

2.1.1. Distancia entre HMMs

Si se tienen dos HMMs λ_1 y λ_2, es posible calcular la distancia de ambos tomando como base la ecuación (2.6), a partir de:

$$d(\lambda_1, \lambda_2) = \lim_{n_\varphi \to \infty} \frac{1}{n_\varphi}\left(\log P(\boldsymbol{\varphi}_1|\lambda_1) - \log P(\boldsymbol{\varphi}_1|\lambda_2)\right) \tag{2.11}$$

La media de distancia definida en (2.11) es no simétrica, una extensión natural de la media anterior, es la media dada por:

$$d(\lambda_1, \lambda_2) = \frac{1}{2}\left(\log(P_{11}P_{22}) - \log(P_{12}P_{21})\right), \tag{2.12}$$

donde,

$$P_{ij} = P(\widehat{\boldsymbol{\varphi}}_i|\lambda_j)^{1/n_{\varphi_i}}, \tag{2.13}$$

siendo n_{φ_i} la longitud de la sucesión $\boldsymbol{\varphi}_i$ generada de forma estocástica a partir del modelo λ_i. La medida $d(\lambda_1, \lambda_2)$ está determinada unívocamente sólo en el límite, cuando $n_{\varphi_i} \to \infty$. Así mismo, se puede demostrar que, si P_{ii} es un máximo global, la distancia de *Kullback-Leibler* tiene las siguientes propiedades [11]:

1. $d(\lambda_1, \lambda_2) = d(\lambda_2, \lambda_1)$
2. $d(\lambda_1, \lambda_2) \geq 0$
3. $d(\lambda_1, \lambda_2) = 0$ si $\lambda_1 \sim \lambda_2$ ó $\boldsymbol{\varphi}_1 = \boldsymbol{\varphi}_2$

Una de las desventajas que tiene este tipo de distancia se encuentra en la necesidad de realizar la simulación, empleando algún método (por ejemplo Monte Carlo), para generar las sucesiones $\widehat{\boldsymbol{\varphi}}_i$, lo cual eleva el costo computacional. En [36], se propone una cota superior para la distancia de Kullback Leibler, que hace innecesario el procedimiento de simulación. Además del problema comentado antes, en [37] se plantea que la métrica de *Kullback-Leibler*, mide la distancia entre probabilidades a través de la entropía relativa y que ésta no es una verdadera métrica.

2.2. Distancia a partir de medidas de similitud

Está medida de similitud estocástica propuesta en [31], fue presentada inicialmente para modelos "discretos", sin embargo, su generalización a modelos continuos se realiza de forma directa. La medida de similitud cuantifica la semejanza entre dos trayectorias estocásticas con dimensión múltiple en correspondencia con los modelos ocultos de Markov dados,

$$s(\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2) = \sqrt{\frac{P_{21}P_{12}}{P_{11}P_{22}}} \tag{2.14}$$

donde,

$$P_{ij} = P(\boldsymbol{\varphi}_i|\lambda_j)^{1/n_{\varphi_i}} \tag{2.15}$$

representa la probabilidad de la sucesión de observación $\boldsymbol{\varphi}_i$ dada por el modelo λ_j, normalizado con respecto a n_{φ_i}, donde n_{φ_i} es la longitud de la sucesión $\boldsymbol{\varphi}_i$. Se puede demostrar [31] que si P_{ii} es un máximo global, la medida de similitud tiene las siguientes propiedades:

1. $s(\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2) = s(\boldsymbol{\varphi}_2, \boldsymbol{\varphi}_1)$
2. $0 < d(\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2) \leq 1$
3. $s(\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2) = 1$ si $\lambda_1 \sim \lambda_2$ ó $\boldsymbol{\varphi}_1 = \boldsymbol{\varphi}_2$

En algunas ocasiones es más conveniente representar la similitud entre dos modelos de Markov a través de una medida de distancia en lugar de una medida de similitud. Dada la medida de similitud $s(\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2)$, la medida de distancia se puede obtener a partir de

$$d(\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2) = -\log s(\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2) \tag{2.16}$$

tal que se cumple $d(\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2) = d(\boldsymbol{\varphi}_2, \boldsymbol{\varphi}_1)$, $d(\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2) \geq 0$ y $d(\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2) = 1$ si $\lambda_1 \sim \lambda_2$ ó $\boldsymbol{\varphi}_1 = \boldsymbol{\varphi}_2$.
A diferencia de las sucesiones de observación $\boldsymbol{\varphi}_i$, las sucesiones de observación $\widehat{\boldsymbol{\varphi}}_i$ de (2.13) no son únicas debido a que son generadas de forma estocástica a partir de $\widehat{\lambda}_i$.
Aunque de forma general $d(\widehat{\lambda}_1, \widehat{\lambda}_2)$ (ecuación (2.12)) y $d(\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2)$ (ecuación (2.16)) no son equivalentes, bajo ciertas presunciones las dos nociones (distancia entre los modelos ocultos y distancia entre sucesiones de observación) convergen a una equivalencia. Específicamente, se tiene que, $d(\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2) = d(\widehat{\lambda}_1, \widehat{\lambda}_2)$ si y sólo si

1. $\lambda_1 \sim \widehat{\lambda}_1$ y $\lambda_2 \sim \widehat{\lambda}_2$

2. P_{11} y P_{22} son máximos globales.

3. $\widehat{n}_{\varphi_i} \to \infty$

Un problema de las medias anteriores, es que no tienen en cuenta la información de la secuencia de estados, seguida por cada una de las secuencias de observaciones en los modelos. Está información tiene gran relevancia para comparar las dinámicas modeladas por ambos modelos [35]. En [27] se presenta un conjunto de métricas entre HMMs, desarrolladas a partir de la definición de probabilidad de *co-emisión* entre dos modelos. Estas medidas tienen en cuenta la información de la secuencia de estados.

2.3. Medidas de distancia a partir de la probabilidad de *co-emisión*

La probabilidad de co-emisión de dos modelos, se define como la probabilidad de que independientemente los modelos generen la misma secuencia sobre un alfabeto υ, se define como [27]:

$$\sum_{\varphi \in \mathfrak{U}} P(\lambda_1|\varphi)P(\lambda_2|\varphi) \tag{2.17}$$

donde $P(\lambda_i|\varphi)$ es la probabilidad de que el modelo λ_i genere la secuencia φ. El problema de está aproximación, es que para calcular la probabilidad de *co-emisión* se requiere que los modelos sean *discretos* y de arquitectura izquierda-derecha [11].
Para calcular la probabilidad de *co-emisión* se construye una tabla indexada por estados de los dos HMM, tal que $\mathbf{A}(\theta, \theta')$ - donde θ es un estado de λ_1 y θ' es un estado de λ_2- contiene la probabilidad de estar en el estado θ en λ_1 y θ' en λ_2 y haber generado independientemente secuencias idénticas en las trayectorias a θ y a θ'. La entrada indexada por los dos estados finales es la probabilidad de *co-emisión*.
La construcción de la tabla A, depende el tipo de estados del modelo. Para estados que emitan símbolos en ambos modelos y con lazos de transición en si mismos, se tiene que:

$$\mathbf{A}(\theta, \theta') = \frac{1}{1-r}\mathbf{A}_0(\theta, \theta') \tag{2.18}$$

donde $A_0(\theta_q, \theta'_q)$ se construye sumando todas las posibles combinaciones de estados con transiciones a θ_q y θ'_q (incluyendo las combinaciones con θ_q o θ'_q pero no ambos) de la siguiente forma:

$$\begin{aligned}\mathbf{A}_0(\theta, \theta') =& p(\mathbf{A}(\theta_i, \theta'_i)\pi_{\theta_i\theta_q}\pi'_{\theta'_i\theta'_q} + A(\theta_j, \theta'_i)\pi_{\theta_j\theta_q}\pi'_{\theta'_i\theta'_q} + \cdots + \mathbf{A}(\theta_q, \theta'_i)\pi_{\theta_q,\theta_q}\pi'_{\theta'_i,\theta'_q} + \ldots \\ &+ \mathbf{A}(\theta_k, \theta'_q)\pi_{\theta_k,\theta_q}\pi'_{\theta'_q,\theta'_q})\end{aligned} \tag{2.19}$$

donde $\pi_{\theta_i\theta_j}$ es la probabilidad de transición del estado i al estado j en el modelo λ y $\pi'_{\theta'_i\theta'_j}$ es la probabilidad de transición del estado θ'_i al estado θ'_j en el modelo λ'. p es la probabilidad de que dos estados independientemente generen el mismo símbolo, así:

$$p = \sum_{\upsilon_k \in \mathfrak{U}} b_{\theta_i}(\upsilon_k)b_{\theta'_i}(\upsilon_k) \tag{2.20}$$

donde $b_{\theta_i}(\varphi_k)$ es la probabilidad de emitir el símbolo υ_k en el estado θ_i del modelo λ y $b_{\theta'_i}(\varphi_k)$ es la probabilidad de emitir el símbolo υ_k en el estado θ'_i del modelo λ'. r en la ecuación (2.18) es la probabilidad de escoger independientemente lazos de transición en si mismos y emitir el mismo símbolo en θ_q y θ'_q, así:

$$r = p\pi_{\theta_q,\theta_q}\pi'_{\theta'_q,\theta'_q} \tag{2.21}$$

La probabilidad de *co-emisión* de dos modelos λ_1 y λ_2 se denota por $P_{\mathbf{A}}(\lambda_1, \lambda_2)$. Basado en la probabilidad de *co-emisión* se definen dos métricas [27]:

$$d_{ang}(\lambda_1, \lambda_2) = \arccos(P_{\mathbf{A}}(\lambda_1, \lambda_2)/\sqrt{P_{\mathbf{A}}(\lambda_1, \lambda_1)P_{\mathbf{A}}(\lambda_2, \lambda_2)}) \tag{2.22}$$

$$d_{dif}(\lambda_1, \lambda_2) = \sqrt{P_{\mathbf{A}}(\lambda_1, \lambda_1) + P_{\mathbf{A}}(\lambda_2, \lambda_2) - 2P_{\mathbf{A}}(\lambda_1, \lambda_2)} \tag{2.23}$$

Las medidas anteriores son derivadas de interpretar un HMM como un vector de secuencias [27]. Así, la probabilidad de *co-emisión* se puede convertir en el producto interno

$$\langle \lambda_1, \lambda_2 \rangle = |\lambda_1|\,|\lambda_2| \cos \upsilon \tag{2.24}$$

de los modelos. Donde υ es el angulo entre los modelos y $|\lambda_i|$ es la longitud de λ_i. El angulo entre los modelos da una medida de ortogonalidad pero no tiene en cuenta la longitud (dos modelos son ortogonales, si y sólo si, no pueden generar secuencias identicas, y paralelo si expresan la misma distribución de probabilidad), mientras que la medida de diferencia es derivada de la medida estándar en espacios de vectores, la norma euclidiana de la diferencia entre los dos vectores.

Aunque las medidas derivadas de la probabilidad de *co-emisión* tienen en cuenta la información de la secuencia de estados de los modelos, no lo hacen a partir de una secuencia de entrada particular; es decir, no se estima la secuencia de estados para una secuencia φ determinada, en cada uno de los modelos, sino que la probabilidad en la cual se basa la métrica, es independiente de una nueva secuencia arbitraria. Una distancia de probabilidad apropiada, debería estar relacionada a una distribución de probabilidad condicional dada una secuencia de observación, en lugar de utilizar distribuciones de probabilidad incondicionales o marginales [38]. Adicionalmente, las métricas derivadas de la probabilidad de *co-emisión*, de igual forma que la medida derivada de la distancia euclídea, se aplican a modelos *discretos* que no pueden ser utilizados en el algoritmo MCE.

2.4. Generalización de distancias a partir de la probabilidad de *Viterbi*

Las medidas descritas en la secciones 2.1 y 2.2 aunque están basadas en probabilidades de secuencias de observación, condicionadas a los modelos comparados, no tienen en cuenta la información de la secuencia de estados seguida por el modelo, para generar la secuencia de observación dada. En [35] se propone utilizar la información de la secuencia de estados más probable en el modelo, a partir de la probabilidad conjunta estado-observación $P(\boldsymbol{\varphi}, \boldsymbol{\theta}|\lambda)$, que puede ser calculada utilizando el algoritmo de *Viterbi* [11], en lugar de la máxima

probabilidad a posteriori (2.13). Ahora, la distancia entre HMMs a partir de la definición (2.11), se convierte en [35]:

$$d\left(\lambda_1,\lambda_2\right)=\lim_{n_\varphi\to\infty}\frac{1}{n_\varphi}\left(\log p\left(\widehat{\boldsymbol{\varphi}}_1,\boldsymbol{\theta}_1|\lambda_1\right)-\log p\left(\widehat{\boldsymbol{\varphi}}_1,\boldsymbol{\theta}_2|\lambda_2\right)\right) \tag{2.25}$$

De igual forma que la ecuación (2.11), la medida definida en (2.25), es no simétrica, una versión simétrica derivada de la ecuación (2.12), está dada por:

$$d\left(\lambda_1,\lambda_2\right)=\frac{1}{2}\left(\log\left(\tilde{P}_{11}\tilde{P}_{22}\right)-\log\left(\tilde{P}_{12}\tilde{P}_{21}\right)\right) \tag{2.26}$$

donde,

$$\tilde{P}_{ij}=P\left(\widehat{\boldsymbol{\varphi}}_i,\boldsymbol{\theta}_j|\lambda_j\right)^{1/n_{\varphi_i}} \tag{2.27}$$

La generalización para el caso de la medida de distancia a partir de similitud, se hace de manera análoga a lo descrito para el caso de la distancia *Kullback-Leibler*. Así, la ecuación (2.15) se convierte en:

$$\tilde{P}_{ij}=P(\boldsymbol{\varphi}_i,\boldsymbol{\theta_j}|\lambda_j)^{1/n_{\varphi_i}} \tag{2.28}$$

Entonces, la distancia a partir de la medida de similitud descrita en la ecuación (2.16), se puede expresar como:

$$\begin{aligned} d\left(\varphi_1,\varphi_2\right) &= -\log\sqrt{\tfrac{\tilde{P}_{21}\tilde{P}_{12}}{\tilde{P}_{11}\tilde{P}_{22}}} \\ &= -\tfrac{1}{2}\left(\log\left(\tilde{P}_{21}\tilde{P}_{12}\right)-\log\left(\tilde{P}_{11}\tilde{P}_{22}\right)\right) \\ &= -\tfrac{1}{2}\left(\left(-\log\tilde{P}_{11}+\log\tilde{P}_{12}\right)+\left(-\log\tilde{P}_{22}+\log\tilde{P}_{21}\right)\right) \\ &= -\tfrac{1}{2}\left(d^*_{12}+d^*_{21}\right) \end{aligned} \tag{2.29}$$

donde,

$$d^*_{12}=-\log\tilde{P}_{11}+\log\tilde{P}_{12}\qquad\text{y}\qquad d^*_{21}=-\log\tilde{P}_{22}+\log\tilde{P}_{21} \tag{2.30}$$

La medida de distancia d^*_{12}, es no simétrica y podría ser interpretada como la distancia que existe entre la secuencia de observación φ_2 y φ_1 con respecto a φ_1 [35].
Si se comparan las medidas descritas en la ecuación (2.30), con la medida de distancia propuesta en la ecuación (1.29), se puede llegar a la conclusión que son iguales para el caso particular en el cual, se tienen únicamente dos clases definidas para el entrenamiento de HMMs por medio del criterio MCE.

Capítulo 3

Extracción de características

3.1. Métodos convencionales de extracción de características

La principal tarea de un extractor de características es seleccionar o combinar las características que contienen más información y remover las componentes redundantes, con el objetivo de mejorar la eficiencia de la etapa subsecuente de clasificación sin degradar su desempeño [39]. La mayor parte de los métodos de extracción de características encontrados en la literatura, están basados en métodos de extracción lineal. La extracción de características lineal, proyecta los vectores de parámetros del espacio paramétrico dentro de un espacio característico, a través de una matriz de transformación lineal $\mathcal{T}$. Suponga que el vector correspondiente a una observación de entrada φ es un vector ρ-dimensional y $\mathcal{T}$ es una matriz $\rho \times \varrho$ ($\rho \geq \varrho$). El vector de características extraído ψ es:

$$\psi = \mathcal{T}^T \varphi \tag{3.1}$$

La diferencia entre los algoritmos de extracción de características, es el criterio por el cual optimizan la transformación $\mathcal{T}$. Los métodos básicos más empleados en tareas de extracción lineal son: El Análisis Discriminante Lineal (*Linear Discriminant Analysis - LDA*)y El Análisis de Componentes Principales (*Principal Component Analysis - PCA*) [3]. Brevemente hablando, LDA optimiza $\mathcal{T}$ maximizando la relación entre la dispersión entre - clases y la dispersión intra - clases; PCA obtiene $\mathcal{T}$ buscando las direcciones en el espacio original que tienen mayor variación.

3.1.1. Análisis de componentes principales

El análisis de componentes principales (*Principal component analysis - PCA*), es una técnica estadística cuyo propósito es condensar la información de un gran conjunto de variables correlacionadas, en otro conjunto con menos variables ("las componentes principales") [3], reteniendo tanto como sea posible la variación presente en el conjunto inicial de datos. Suponga que φ es un vector aleatorio ϱ-dimensional. PCA primero busca una función lineal $\alpha_1^T \varphi$ de φ que tenga máxima varianza, donde $\alpha_1 = \{\alpha_{11}, \alpha_{12}, ..., \alpha_{1m}\}$ es un vector

ϱ-dimensional y

$$\alpha_1^T\varphi = \alpha_{11}\varphi_1 + \alpha_{12}\varphi_2 + ... + \alpha_{1\varrho}\varphi_\varrho = \sum_{i=1}^{\varrho}\alpha_{1i}\varphi_i \tag{3.2}$$

Luego busca una segunda función lineal $\alpha_2^T\varphi$ que es no correlacionada con $\alpha_1^T\varphi$ y tiene la segunda varianza máxima. Este procedimiento se repite hasta que la k-ésima función deseada $\alpha_k^T\varphi$ sea encontrada [39]. Estas k variables, $\alpha_1^T\varphi, \alpha_2^T\varphi, ..., \alpha_k^T\varphi$, son llamadas las componentes principales y en general pueden ser encontradas hasta ϱ componentes principales.

El primer componente principal se define como la combinación lineal de variables originales que tienen varianza máxima. Matemáticamente considere el primer componente $\alpha_1^T\varphi$, α_1 maximiza var $\left[\alpha_1^T\varphi\right] = \alpha_1^T\Sigma\alpha_1$ sujeto a $\alpha_1^T\alpha_1 = 1$. Donde Σ es la matriz de covarianza de las observaciones. Usando multiplicadores de Lagrange se tiene:

$$\alpha_1^T\Sigma\alpha_1 - u_1(\alpha_1^T\alpha_1 - 1) \tag{3.3}$$

donde, u_1 es un multiplicador de Lagrange. Derivando (3.3) con respecto a α_1 se tiene:

$$(\Sigma - u_1 I_\varrho)\,\alpha_1 = 0 \tag{3.4}$$

donde I_ϱ es una matriz identidad de tamaño $\varrho \times \varrho$. Note que la cantidad a ser maximizada es:

$$\alpha_1^T\Sigma\alpha_1 = \alpha_1^T u_1\alpha_1 = u_1\alpha_1^T\alpha_1 = u_1 \tag{3.5}$$

Lo que implica que u_1 es el mayor valor propio de la matriz Σ y α_1 su correspondiente vector propio [39, 40].

Considere la segunda componente principal, $\alpha_2^T\varphi$, maximizar $\alpha_2^T\Sigma\alpha_2$, sujeto a que sea no correlacionado con la primera componente principal y $\alpha_2^T\alpha_2 = 1$. Esto es,

$$\alpha_2^T\Sigma\alpha_2 - u_2\left(\alpha_2^T\alpha_2 - 1\right) - \phi\alpha_2^T\alpha_1 \tag{3.6}$$

donde u_2 y ϕ son multiplicadores de Lagrange. Derivando 3.6 con respecto a α_2 se tiene:

$$\Sigma\alpha_2 - u_2\alpha_2 - \phi\alpha_1 = 0 \tag{3.7}$$

multiplicando el lado izquierdo de (3.7) por α_1^T, se tiene:

$$\alpha_1^T\Sigma\alpha_2 - u_2\alpha_1^T\alpha_2 - \phi\alpha_1^T\alpha_1 = 0 \tag{3.8}$$

en la ecuación (3.8), los primeros dos términos son iguales a cero, razón por la cual, para que se cumpla la igualdad es necesario que $\phi = 0$. Así, la ecuación (3.7) se convierte en:

$$\Sigma\alpha_2 - u_2\alpha_2 = 0 \tag{3.9}$$

Una vez más, $u_2 = \alpha_2^T\Sigma\alpha_2$, por consiguiente u_2 es el segundo más grande valor propio de la matriz Σ y α_2 su correspondiente vector propio. Siguiendo la misma estrategia puede ser mostrado [39], que el vector de coeficientes α_k de la k-ésima componente principal, es el vector propio correspondiente al k-ésimo mayor valor propio de Σ.

PCA para reducción de dimensionalidad en clasificación

Para un conjunto $\mathcal{J}$ de datos ϱ-dimensionales, los ρ ejes principales $T_1, T_2, ..., T_\rho$, donde $1 \leq \rho \leq \varrho$ son ejes ortonormales, sobre los cuales la varianza retenida es máxima en el espacio proyectado. Generalmente, $T_1, T_2, ..., T_\rho$ pueden ser dados por los ρ vectores propios asociados a los mayores valores propios de la matriz de covarianza Σ de la muestra, dada por:

$$\Sigma = \frac{1}{N} \sum_{i=1}^{N} (\varphi_i - \mu)^T (\varphi_i - \mu) \tag{3.10}$$

donde $\varphi_i \in \mathcal{J}$, μ es la media de la muestra y N es el número de muestras, tal que:

$$\Sigma T_i = u_i T_i \quad i \in 1, ..., \varrho \tag{3.11}$$

Los ϱ componentes principales de un vector de observaciones dado $\varphi \in \mathcal{J}$ están dados por:

$$\psi = [\psi_1, ..., \psi_\rho] = \left[T_1^T \varphi, ..., T_\rho^T \varphi\right] = \mathcal{T}^T \varphi \tag{3.12}$$

Los ϱ componentes principales de φ, son entonces no correlacionados en el espacio proyectado. En un problema multi-clase, las variaciones de los datos son determinadas en una base global [41], que significa que los ejes principales son derivados de una matriz de covarianza global. Una suposición hecha por la reducción de dimensión por PCA es que la mayor parte de la información contenida en los vectores de observación es contenida en el subespacio generado por los primeros ρ ejes principales, donde $\rho < \varrho$. Por consiguiente, cada vector de datos original puede ser representado por su vector de componentes principales:

$$\psi = \mathcal{T}^T \varphi \tag{3.13}$$

donde $\mathcal{T}$ es una matriz $\varrho \times \rho$.
El mérito de PCA está en que las variables extraídas tiene la mínima correlación a lo largo de los ejes principales. Sin embargo, existen algunos defectos que se encuentran en PCA. Primero, cómo se menciona en [40], PCA es un método sensitivo a la escala, es decir, las componentes principales pueden ser dominadas por elementos con grandes varianzas. Otro problema que presenta PCA es que las direcciones de máxima varianza no son necesariamente las direcciones de máxima discriminación dado que no utiliza la información de etiquetado de las clases.

3.1.2. Análisis discriminate lineal

Discriminate lineal de Fisher

El objetivo del discriminante lineal de Fisher es separar la clases, proyectando las muestras de las clases de un espacio ϱ-dimensional dentro de una linea. Para un problema que envuelve K clases, la generalización natural del discriminante lineal de Fisher envuelve $K-1$ funciones discriminantes. Así, la proyección es de un espacio ϱ-dimensional a un espacio $K-1$-dimensional. Es tácitamente asumido que $\varrho \geq K$ [3].
Suponga que se tienen K clases, $C_1, ..., C_K$, Sea φ_{ji} el i-ésimo vector de observaciones de

la clase C_j, donde $i = 1, ..., N_j$ y N_j es el número de observaciones de la clase j. El vector de medias y la matriz de covarianza de la clase j están dados por:

$$\mu_j = \frac{1}{N_j} \sum_{i=1}^{N_j} \varphi_{ji} \tag{3.14}$$

y

$$\Sigma_j = \frac{1}{N_j} \sum_{i=1}^{N} (\varphi_{ji} - \mu_j)(\varphi_{ji} - \mu_j)^T \tag{3.15}$$

La matriz de covarianza intra-clase Σ_W está dada por:

$$\Sigma_W = \sum_{j=1}^{K} \Sigma_j \tag{3.16}$$

La media y la matriz de covarianza para el conjunto total de datos Σ_T, están dadas por:

$$\mu = \frac{1}{N} \sum_{j=1}^{K} \sum_{i=1}^{N_j} \varphi_{ji} = \frac{1}{N} \sum_{j=1}^{K} N_j \mu_j \tag{3.17}$$

y

$$\Sigma_T = \sum_{j=1}^{K} \sum_{i=1}^{N_j} (\varphi_{ji} - \mu)(\varphi_{ji} - \mu)^T \tag{3.18}$$

donde $N = \sum_{j=1}^{K} N_j$. Se sigue entonces que:

$$\begin{aligned} \Sigma_T &= \sum_{j=1}^{K} \sum_{i=1}^{N_j} (\varphi_{ji} - \mu_j + \mu_j - \mu)(\varphi_{ji} - \mu_j + \mu_j - \mu)^T \\ &= \sum_{j=1}^{K} \sum_{i=1}^{N_j} (\varphi_{ji} - \mu_j)(\varphi_{ji} - \mu_j)^T + \sum_{j=1}^{K} \sum_{i=1}^{N_j} (\mu_j - \mu)(\mu_j - \mu)^T \\ &= \Sigma_W + \sum_{j=1}^{K} N_j (\mu_j - \mu)(\mu_j - \mu)^T \end{aligned} \tag{3.19}$$

El segundo término en la ecuación (3.19) se define como la matriz de covarianza entre - clases. Se tiene entonces que:

$$\Sigma_B = \sum_{j=1}^{K} N_j (\mu_j - \mu)(\mu_j - \mu)^T \tag{3.20}$$

y

$$\Sigma_T = \Sigma_W + \Sigma_B \tag{3.21}$$

En este caso, la proyección de un espacio ϱ-dimensional a un espacio ρ-dimensional es realizado por $K - 1$ funciones discriminantes:

$$\psi_i = \boldsymbol{w}_i^T \varphi \qquad i = 1, 2, ..., K - 1 \tag{3.22}$$

La proyección en la ecuación (3.22), puede ser reescrita en forma matricial como:

$$\psi = \boldsymbol{W}^T \varphi \tag{3.23}$$

Las muestras $\varphi_1, ..., \varphi_N$ son proyectadas a un conjunto correspondiente de muestras $\psi_1, ..., \psi_N$, las cuales pueden ser descritas por su propio vector de medias y matriz de covarianza, definidas por:

$$\tilde{\mu}_j = \frac{1}{N_j} \sum_{i=1}^{N_j} \varphi_{ji} \tag{3.24}$$

$$\tilde{\mu} = \frac{1}{N} \sum_{i=1}^{N_j} N_j \tilde{\mu}_j \tag{3.25}$$

$$\tilde{\Sigma}_W = \sum_{j=i}^{K} \sum_{i=1}^{N_j} (\psi_{ji} - \tilde{\mu}_j) (\psi_{ji} - \tilde{\mu}_j)^T \tag{3.26}$$

y

$$\tilde{\Sigma}_B = \sum_{j=i}^{K} N_j (\tilde{\mu}_j - \tilde{\mu}) (\tilde{\mu}_j - \tilde{\mu})^T \tag{3.27}$$

De una forma directa se puede mostrar que [39]:

$$\tilde{\Sigma}_W = W^T \Sigma_W W \tag{3.28}$$

y

$$\tilde{\Sigma}_B = W^T \Sigma_B W \tag{3.29}$$

El discriminante lineal de Fisher es entonces definido como una función lineal $W^T \varphi$ para la cual la función criterio

$$J(W) = \frac{\left|\tilde{\Sigma}_B\right|}{\left|\tilde{\Sigma}_W\right|} = \frac{W^T \Sigma_B W}{W^T \Sigma_W W} \tag{3.30}$$

es máximo.

Se puede mostrar que la solución de la ecuación (3.30), corresponde a calcular los mayores $k-1$ valores propios generalizados (y sus correspondientes vectores propios) de la matriz $\Sigma_W^{-1} \Sigma_B$.

Aunque la evidencia practica ha mostrado que el análisis discriminante es efectivo, una separación significante no implica necesariamente buena clasificación [42]. El análisis discriminante multi-clase está relacionado con la búsqueda de una transformación lineal, que reduzca la dimensión de un modelo estadístico ϱ-dimensional de K clases, a un espacio de representación de dimensión $K-1$, manteniendo el máximo conjunto de información discriminante en un modelo de baja dimensión. Se ha mostrado que para un problema multi-clase [42], el criterio de Fisher está maximizando la distancia cuadrática media entre las clases en un espacio de baja dimensión, lo que es claramente diferente de minimizar el error de clasificación [43]. Cuando se maximiza la distancia cuadrática media, el par de clases que tienen mayor distancia, dominan completamente la descomposición en valores

propios. La transformación resultante preserva la distancia de las ya bien separadas clases. Como consecuencia, puede existir un gran solapamiento de las clases restantes, lo que recae en un clasificación subóptima con baja tasa de exactitud.
Para el caso en el cual se cuenta únicamente con dos clases, la proyección del conjunto de datos de entrada a una única línea, no entrega suficiente información discriminante en un problema medianamente complejo.

3.2. Extracción de características para mínimo error de clasificación

El método de extracción de características basado en el algoritmo de mínimo error de clasificación [44], pretende de igual forma que los métodos convencionales de extracción de características, proyectar los datos de entrada a un espacio de menor dimensión, mediante una transformación lineal. En este caso, las funciones discriminantes son medidas basadas en la distancia de Mahalanobis, dadas por [1]:

$$g_i(\varphi, \Lambda) = (\varphi - \mu^{(i)})^T \left(\Sigma^{(i)}\right)^{-1} (\varphi - \mu^{(i)}) \tag{3.31}$$

donde μ es la media de la clase y Σ la matriz de covarianza. Para poder utilizar estas funciones discriminantes dentro del algoritmo MCE, se define una medida de mala clasificación en un problema multi-clase dada por:

$$d_k(\varphi, \Lambda) = \frac{\left(\frac{1}{N-1} \sum_{\forall\, i \neq k} g_i(\varphi, \Lambda_i)^\eta\right)^{1/\eta}}{g_k(\varphi, \Lambda_k)} \tag{3.32}$$

La medida de mala clasificación de la ecuación (3.32), está embebida en la función sigmoidal definida en la ecuación (1.32). La medida (3.32) es equivalente a la medida definida en (1.29), pero para el caso en el cual las funciones discriminantes no son de tipo logaritmico.
Debido a que el método reduce el espacio de características proyectando el vector de entrada a través de una transformación lineal, se asume que las características a la entrada son de tipo estático.
La matriz de transformación $\mathbf{W}$ es estimada utilizando el criterio de optimización del algoritmo MCE, razón por la cual la transformación está orientada a disminuir el error de clasificación y actúa por lo tanto de manera conjunta con el objetivo de diseño del clasificador.
Si la transformación del espacio de entrada se expresa de igual forma como en la ecuación (3.23), la función de perdida a ser minimizada puede ser expresada como:

$$\ell(\psi) = \frac{1}{1 + \exp\left(-\gamma d\left(\mathbf{W}^T\varphi, \Lambda\right)\right)} = \frac{1}{1 + \exp\left(-\gamma d\left(\psi, \Lambda\right)\right)} \tag{3.33}$$

Los elementos de la transformación lineal pueden ser optimizados empleando la regla (1.34), como:

$$\mathbf{W}_{sq}(n+1) = \mathbf{W}_{sq}(n) - \varepsilon \left.\frac{\partial \ell_i}{\partial \mathbf{W}_{sq}}\right|_{\mathbf{W}_{sq}=\mathbf{W}_{sq}(n)} \tag{3.34}$$

donde s y q son los indicadores de fila y columna de la matriz de transformación y ℓ_i es la perdida empírica para el conjunto completo de datos.
Para un caso bi-clase, el gradiente de la ecuación (3.34) con respecto a $\mathbf{W}$ puede ser calculado por [1]:

– *MCE Convencional*:

$$\frac{\partial \ell_i}{\partial \mathbf{W}_{sq}} = \varepsilon \ell_i \left(1 - \ell_i\right) \left(\frac{\partial g_k \left(\mathbf{W}\varphi, \Lambda\right)}{\partial \mathbf{W}_{sq}} - \frac{\partial g_j \left(\mathbf{W}\varphi, \Lambda\right)}{\partial \mathbf{W}_{sq}} \right) \tag{3.35}$$

– *MCE Alternativo*:

$$\begin{aligned} \frac{\partial \ell_i}{\partial \mathbf{W}_{sq}} &= \varepsilon \ell_i \left(1 - \ell_i\right) \\ &\left(\frac{\left(\partial g_j(\mathbf{W}\varphi,\Lambda)/\partial \mathbf{W}_{sq}\right) g_k(\mathbf{W}\varphi,\Lambda) - \left(\partial g_k(\mathbf{W}\varphi,\Lambda)/\partial \mathbf{W}_{sq}\right) g_j(\mathbf{W}\varphi,\Lambda)}{\left(g_k(\mathbf{W}\varphi,\Lambda)\right)^2} \right) \end{aligned} \tag{3.36}$$

dado que en este caso las funciones discriminantes están basadas en la distancia de Mahalanobis, las derivas parciales de las funciones g son de la forma:

$$\frac{\partial g_m \left(\mathbf{W}\varphi, \Lambda\right)}{\partial \mathbf{W}_{sq}} = \left(\mathbf{W}\varphi - \mu\right)^T \Sigma^{-1} \left(\mathbf{W}\varphi - \mu\right) \tag{3.37}$$

Un aspecto importante que debe ser considerado en relación con el método de reducción de dimension empleando el algoritmo MCE, es la inicialización de los parámetros, debido a que el método de gradiente descendente utilizado por el algoritmo MCE no garantiza el hallazgo de un mínimo global de la función de pérdida.
Uno de los problemas que presenta el método descrito anteriormente, es que de manera similar a los métodos convencionales de extracción de características, no está desarrollado para espacios de representación que utilizan variables dinámicas, ya que desconoce la información dinámica que éstas presentan. Por otro lado, si se observa detenidamente, a diferencia de otros métodos como PCA, la transformación fue optimizada sin ninguna restricción, lo que conlleva a que las características en el espacio transformado no están acotadas. Este hecho puede presentar diversos problemas de sesgo en la clasificación, debido a la influencia negativa que tiene la diferencia entre ordenes de magnitud de los parámetros considerados, por ejemplo, cuando se trabaja con base en el cálculo de matrices de covarianza [45].

Parte III

Marco experimental

Capítulo 4

Análisis experimental de la reducción de espacios empleando HMM

4.1. Extracción de características y entrenamiento simultaneo de HMM

De manera similar al método de EC descrito en la sección 3.2, se puede extender el algoritmo MCE orientado a la estimación de parámetros de un HMM, a un método de EC que emplee el mismo criterio de optimización. De está manera, se elimina el problema de inconsistencia entre las etapas de EC y clasificación expuesto en la sección 3.1. Además, el método de EC derivado del algoritmo MCE para HMMs, tiene en cuenta la información de la dinámica temporal de las características.
El método de EC para mínimo error de clasificación, estima una transformación lineal utilizando el criterio MCE. Sin embargo, en el caso en que las características con las que se cuenta, están estructuradas en forma de secuencias de datos temporales, utilizar una única matriz de transformación, desconoce el comportamiento dinámico-estocástico del proceso modelado. Una forma directa de tomar en cuenta el comportamiento dinámico del proceso en la EC, es utilizar el modelo dinámico proporcionado por la secuencia de estados del HMM, y estimar un modelo de transformación del espacio no lineal, basado en esta dinámica. Se pueden entonces, entrenar una matriz de transformación para cada uno de los estados del modelo (dependiente de los estados). La transformación del espacio original se realiza de la siguiente manera: dada una secuencia de observaciones y estimada una secuencia de estados más probable en el modelo $\bar{\theta}$ (debido a que los parámetros del HMM en el algoritmo descrito en la sección 1.3.2, se optimizan teniendo en cuenta dicha secuencia de estados), si la secuencia de observación en el tiempo t, se encuentra en el estado θ_t, la observación t de la secuencia, será transformada utilizando la transformación asociada al estado θ_t. Formalmente, existe una transformación $\mathcal{W} = \{\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_{n_\vartheta}\}$, que es en forma general, el conjunto de transformaciones relacionadas con cada estado del modelo.
Es posible considerar dos configuraciones diferentes para obtener la estimación del modelo de transformación. La primera configuración consiste en obtener un único modelo de transformación para todas las clases, esta configuración en particular es similar a la

desarrollada en [1] (pero adicionalmente considera la información de la dinámica). De esta manera, las matrices de transformación por estado, son compartidas entre los estados de los modelos HMM pertenecientes a clases diferentes. En la segunda configuración, se obtiene una transformación por cada uno de los modelos (o clases). Es decir, para cada estado en cada uno de los modelos, es estimada una transformación.
La estimación de la transformación derivada de la ecuación (1.34), para el estado j en la primera configuración, está dada por:

$$\mathbf{W}_j(n+1) = \mathbf{W}_j(n) - \varepsilon \left. \frac{\partial \ell_i(\xi;\lambda)}{\partial \mathbf{W}_j} \right|_{\lambda=\lambda_n} \tag{4.1}$$

donde $\xi = \mathcal{W}^T\varphi$ es la secuencia de observación transformada por $\mathcal{W}$. La función ℓ_i corresponde a la función sigmoidal (función de pérdida) definida en la ecuación (1.32). Para completar la definición de la función de pérdida, es necesario establecer la medida de *mala clasificación* d_i en función de la cual, está definida la ecuación (1.32).
Debido a que el problema abordado en este trabajo es un problema bi-clase (clasificación entre normal y patológico), se utilizará la distancia de similitud no simétrica, definida en la ecuación (2.30), debido a que de todas las distancias estudiadas en el capítulo 2, está medida es la única que permite hacer la actualización de la transformación, a partir del algoritmo GPD, empleando en cada iteración la información de una única secuencia de observación.
El problema de estimación de la transformación $\mathcal{W}$, para la primera configuración siguiendo la ecuación (4.1), se convierte por regla de la cadena en:

$$\begin{aligned} \frac{\partial \ell_i(\xi;\lambda)}{\partial \mathbf{W}_j} &= \frac{\partial \ell_i}{\partial d_i}\frac{\partial d_i}{\partial \mathbf{W}_j} \\ &= \frac{\partial \ell_i}{\partial d_i}\left(-\frac{\partial g_i(\xi;\lambda)}{\partial \mathbf{W}_j} + \frac{\partial g_k(\xi;\lambda)}{\partial \mathbf{W}_j}\right) \end{aligned} \tag{4.2}$$

De la ecuación (4.2) es fácilmente deducible, que para la segunda configuración, debido a que la transformación es asociada a una clase, uno de los dos factores de la ecuación (4.2) se hace igual a cero. Por lo tanto en la segunda configuración, la estimación de la transformación asociada al modelo de la clase i en el estado j, puede ser derivada de (1.34) como:

$$\mathbf{W}_j^{(i)}(n+1) = \mathbf{W}_j^{(i)}(n) - \varepsilon \left. \frac{\partial \ell_i(\xi;\lambda)}{\partial \mathbf{W}_j^{(i)}} \right|_{\lambda=\lambda_n} \tag{4.3}$$

De igual forma a como se establece la restricción en la transformación para el caso de PCA (ecuación (3.3)), debe establecerse una restricción a la transformación $\mathcal{W}$, de tal forma que los datos en el espacio transformados estén acotados (debido a los problemas expuestos en la sección previa). Para esto es necesario que el operador de transformación sea un operador acotado [46], es decir, que la transformación realizada por el operador $\mathcal{W}$ sea del tipo $\mathcal{W}: \mathcal{H} \to \mathcal{H}$. Es posible establecer una restricción para la transformación $\mathcal{W}$, limitando la norma del vector transformado a 1.
Un aspecto que debe ser tenido en cuenta, es la manera en la cual debe ser implementada la restricción en el algoritmo de re-estimación, dado que el algoritmo GPD, es un método de optimización sin restricciones. Podría entonces pensarse en una implementación de la

restricción, de la misma forma cómo se restringen los parámetros de los HMM para su estimación por MCE [10]. Sin embargo, en este caso las restricciones no se derivan de la necesidad de satisfacer condiciones probabilísticas, sino de la necesidad de mantener acotados los parámetros transformados. Una posible restricción está dada por:

$$\mathbf{W} = \frac{\tilde{\mathbf{W}}}{\left\|\tilde{\mathbf{W}}\varphi\right\|} \tag{4.4}$$

El vector transformado dado por:

$$\mathbf{W}^{\mathbf{T}}\varphi = \frac{\tilde{\mathbf{W}}^{\mathbf{T}}\varphi}{\left\|\tilde{\mathbf{W}}\varphi\right\|} \tag{4.5}$$

será entonces de norma 1. Para poder ser incluido en el algoritmo GPD, la función que describe al parámetro restringido debe ser derivable. Además, la función a derivar debe corresponder a un campo escalar, debido a que la derivada de un campo escalar con respecto a un vector, es un campo vectorial, mientras que la derivada de un campo vectorial con respecto a un vector, da como resultado un super-vector [46], lo que impide su implementación en el algoritmo. En este caso, la definición del parámetro restringido dada en (4.4), es una función derivable, pero no es un campo escalar, lo que impide su uso.
Una forma alternativa de incluir la restricción en el algoritmo de re-estimación, como se muestra en [47], hace necesaria la definición de una nueva función de optimización, que para el caso de un vector de características estático esta dada por:

$$f_i = \ell_i\left(\xi_n;\lambda\right) - \kappa\left(\left\|\left(\mathbf{W}^{(i)}\right)^T\varphi_n\right\| - 1\right) \tag{4.6}$$

Donde κ es constante. Para el caso en el cual se tienen una secuencia de observaciones en lugar de una única observación, la función f generalizada esta dada por:

$$f_i\left(\xi_n;\lambda\right) = \ell_i\left(\xi_n;\lambda\right) - \kappa\left(\frac{1}{n_\varphi}\sum_{t=1}^{n_\varphi}\left(\left\|\left(\mathbf{W}_{\bar{q}_t}^{(i)}\right)^T\varphi_t\right\| - 1\right)\right) \tag{4.7}$$

Tomando como base la ecuación (4.7), la formula de re-estimación de la transformación dada en (4.3), se convierte en:

$$\mathbf{W}_j^{(i)}(n+1) = \mathbf{W}_j^{(i)}(n) - \varepsilon\left.\frac{\partial f_i\left(\xi;\lambda\right)}{\partial \mathbf{W}_j^{(i)}}\right|_{\lambda=\lambda_n} \tag{4.8}$$

donde,

$$\frac{\partial f_i\left(\xi;\lambda\right)}{\partial \mathbf{W}_j^{(i)}} = \frac{\partial \ell_i\left(\xi;\lambda\right)}{\partial \mathbf{W}_j^{(i)}} - \kappa\left(\frac{1}{n_\varphi}\sum_{t=1}^{n_\varphi}\delta\left(\bar{\theta}_t - j\right)\frac{\partial}{\partial \mathbf{W}_j^{(i)}}\left(\left\|\left(\mathbf{W}_j^{(i)}\right)^T\varphi_t\right\| - 1\right)\right) \tag{4.9}$$

$$\frac{\partial \ell\left(\xi;\lambda\right)}{\partial \mathbf{W}_j^{(i)}} = \gamma\ell\left(d\right)\left(1-\ell\left(d\right)\right)\frac{\partial g_i\left(\xi;\lambda\right)}{\partial \mathbf{W}_j^{(i)}} \tag{4.10}$$

$$\frac{\partial g_i(\xi;\lambda)}{\partial \mathbf{W}_j^{(i)}} = \frac{1}{n_\varphi}\sum_{t=1}^{n_\varphi}\delta\left(\bar{\theta}_t - j\right)\left(b_j^{(i)}\left(\xi_t\right)\right)^{-1}\sum_{r=1}^{M} c_{jr}^{(i)}\left|\Sigma_{jr}^{(i)}\right|^{-1/2}(2\pi)^{-\rho/2}\frac{\partial b_j^{(i)}\left(\mathbf{W}_j^{(i)}\varphi_t\right)}{\partial \mathbf{W}_j^{(i)}} \tag{4.11}$$

$$\frac{\partial b_j^{(i)}\left(\mathbf{W}_j^{(i)}\varphi_t\right)}{\partial \mathbf{W}_j^{(i)}} = -\exp\left(-\tfrac{1}{2}\left(\mathbf{W}_j^{(i)}\varphi_t - \mu_{jr}^{(i)}\right)^T\Sigma_{jr}^{(i)}\left(\mathbf{W}_j^{(i)}\varphi_t - \mu_{jr}^{(i)}\right)\right)\left(\Sigma_{jr}^{(i)}\right)^{-1}\ldots \left(\mathbf{W}_j^{(i)}\varphi_t - \mu_{jr}^{(i)}\right)\varphi_t^T \tag{4.12}$$

La derivada de la restricción está dada por:

$$\frac{\partial}{\partial \mathbf{W}_j^{(i)}}\left(\left\|\left(\mathbf{W}_j^{(i)}\right)^T\varphi_t\right\| - 1\right) = \left(\varphi_t^T\mathbf{W}_j^{(i)}\left(\mathbf{W}_j^{(i)}\right)^T\varphi_t\right)^{-1/2}\varphi_t\varphi_t^T\mathbf{W}_j^{(i)} \tag{4.13}$$

De esta manera, cuando se calcula la secuencia de estados más probable, es posible transformar la secuencia de observaciones, asociando cada observación a un estado y transformándola. La ventaja de esta aproximación se encuentra en que la transformación $\mathcal{W}$, tiene en cuenta la información dinámica del proceso y es estimada a partir de la maximización de la distancia definida en (2.30).

Capítulo 5

Esquema de trabajo

5.1. Descripción de las bases de datos

Base de datos de señales de voz – BD1

Esta base de datos pertenece al Grupo de Control y Procesamiento Digital de Señales de la Universidad Nacional de Colombia sede Manizales, contiene 90 registros de pronunciaciones de la vocal sostenida $/a/$, repartidas de forma no balanceada (ver Tabla 5.1), de los cuales 40 corresponden a pacientes con voz normal y 50 pacientes con voz disfónica. Los registros fueron adquiridos con una frecuencia de muestreo de 22050 Hz y una duración aproximada de 2,3 *seg*. La tabla 5.1, resume el

Tabla 5.1: Número de muestras en la base de datos BD1

Grupos de observaciones	Número de registros
Patológicas	50
Normales	40

Base de datos de señales de voz – BD2

Esta base de datos fue desarrollada por el Massachusetts Eye and Ear Infirmary [48]. Debido a la heterogeneidad de la base de datos (diferente frecuencia de muestreo en la adquisición de los registros), los registros utilizados fueron re-muestreados a una frecuencia de muestreo de 25 kHz y con una resolución de 16 $bits$. Corresponden a pronunciaciones de la vocal sostenida $/ah/$. Se utilizaron 173 registros de pacientes patológicos (con una amplia gama de patologías vocales orgánicas, neurológicas, traumáticas y psíquicas) y 53 registros de pacientes normales (ver Tabla 5.2), de acuerdo con los registros enumerados en [49]. Los registros de pacientes patológicos tienen una duración aproximada de 1 s, mientras que en los registros de pacientes normales la duración es de unos 3 s. Este hecho, permite equilibrar el número de vectores de características de cada clase y no sesgar el entrenamiento hacia una clase en particular. Por otro lado, debido a que el número de voces de la clase patológica es mayor, la muestra de la clase patológica es estadísticamente más representativa de la población, y por tal motivo se puede obtener un mejor modelado inter sujeto de dicha clase. Este hecho no implica un sesgo del sistema hacia la clase patológica,

debido a que típicamente, en el reconocimiento de patologías de voz, la dispersion en el espacio de características de la clase patológica es mucho mayor que el de la clase normal.

Tabla 5.2: Número de muestras en la base de datos BD2

Grupos de observaciones	Número de registros
Patológicas	173
Normales	53

5.2. Parametrización

Debido a que el método propuesto será probado en un problema de detección de patologías de voz, las medidas empleados, fueron escogidas cómo las más empleadas para este problema específico.
De acuerdo con el modelo usual de la voz [50], ésta está compuesta de una secuencia de excitación en convolución con la respuesta al impulso del sistema vocal. Para modelar esta respuesta, en las tareas de procesamiento de señales de voz es común el empleo de los coeficientes derivados del análisis de predicción lineal *Linear Predictive Coefficients (LPC)* y de los *Mel-Frequency Cepstrum Coefficients (MFCC)* [51]. Los MFCCs pueden estimarse usando una aproximación paramétrica derivada de los LPC o de manera no paramétrica basados en la Transformada rápida de Fourier (*Fast Fourier Transform - FFT*). Sin embargo, la aproximación no paramétrica permite modelar los efectos de las patologías en la excitación (pliegues vocales) y en el sistema (tracto vocal), mientras que el enfoque paramétrico presenta problemas debido a que las patologías introducen no linealidades en el modelo [52]. Por tal motivo, en este trabajo se emplean los coeficientes MFCC derivados del cálculo de la FFT.
La escala de frecuencias *mel*, en la que esta basada la representación perceptual de los MFCC, es una unidad de medida de la frecuencia percibida y no corresponde linealmente a la frecuencia física de la señal, debido a que el sistema auditivo humano aparentemente no percibe las frecuencias de manera lineal [50]. Esta medida está relacionada con el hecho de que un especialista en patologías de voz con suficiente experiencia, puede detectar la presencia de anormalidad en la voz a partir de la audición de la misma.
Además de los MFCC, se han considerado, dentro de los vectores de características, parámetros relacionados con mediciones de ruido, diseñados para medir la componente de ruido relativo en las señales de voz. En particular se utilizó la relación armónico ruido (*Harmonic-to-Noise Ratio - HNR*) [53], la energía de ruido normalizada (*Normalized Noise Energy - NNE*) [54] y la relación excitación glottal ruido (*Glottal to Noise Excitation Ratio*) [55], debido a que estas medidas dan una idea de la calidad y grado de normalidad de la voz.
El vector de características ϱ-dimensional se forma concatenando el conjunto de parámetros de ruido mensionados, además de su primera derivada temporal debido, a que la velocidad de los cambios en los coeficientes dan información importante de su comportamiento dinámico [52]. La figura 5.1 muestra gráficamente la composición del vector de

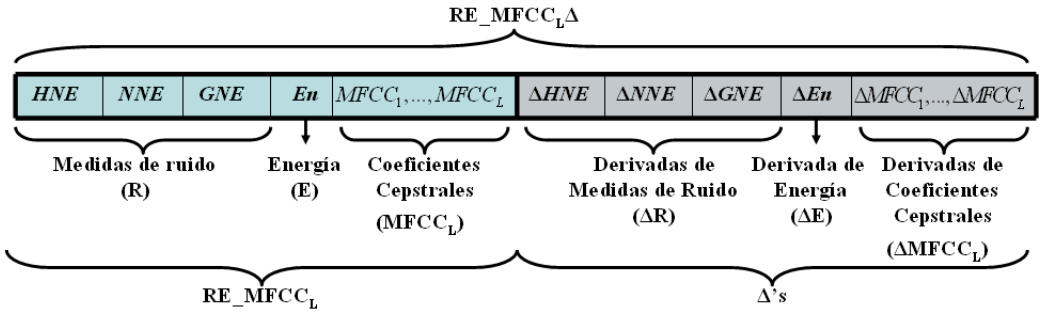


Figura 5.1: Parámetros contenidos en el vector de características.

características empleado. En la figura 5.1, el parámetro En es la energía medida por trama de la señal. El número de coeficientes $MFCC$ utilizados en el vector, está dado por L, que para las pruebas realizadas es igual a 12. Δ es el conjunto de derivadas de cada uno de los parámetros anteriores. El cálculo de Δ fue realizado por medio de un filtro FIR anti-simétrico de respuesta al impulso finita y de longitud 9, para evitar la distorsión de fase de la secuencia temporal [56].

5.3. Selección de variables dinámicas

La variabilidad presente en el conjunto de características considerado, puede ser asociada a la cantidad de información que dicho conjunto contiene.
Es posible plantear un criterio de selección, que permita la identificación de aquellas variables que más peso o relevancia aportan a la variabilidad total, examinando el nivel de correlación del conjunto de características dinámicas con respecto a las componentes que maximizan la variabilidad [57]. Debido a que la magnitud absoluta de los vectores propios ponderados por sus respectivos valores propios, determinan el nivel de correlación entre las variables originales y las componentes principales, se pueden identificar como variables relevantes aquellas asociadas a las mayores magnitudes absolutas anteriormente mencionadas [58].
El conjunto de variables dinámicas obtenidas en la etapa de parametrización, fue reducido empleando una metodología de selección que hace uso del criterio antes mencionado. Una explicación amplia de la metodología puede ser encontrada en el apéndice C.

5.4. Toma de decisión

Cuando se emplean HMMs como clasificadores, la asignación de una nueva muestra (secuencia de observación) a una clase, típicamente se realiza calculando la probabilidad de que cada modelo genere la secuencia de observación dada. La muestra es asignada a la clase del modelo que proporcionó la mayor probabilidad.
A partir del teorema de decisión de Bayes, es posible calcular una puntuación (o *score*) para cada una de las muestras que permita estimar un umbral de decisión óptimo. La puntuación para el caso de HMMs, puede ser calculada como el logaritmo del cociente

entre las probabilidades de generación de la muestra de ambos modelos, conocido como *razón de verosimilitud*. A partir del conjunto de puntuaciones de las muestras de entrenamiento, se construyen las curvas de distribución de puntuaciones verdaderas (puntuaciones de muestras de la case 1) y puntuaciones falsas (puntuaciones de muestras de la clase 0). Así, se puede calcular un umbral de decisión de tal manera que el error de clasificación sea mínimo. En la figura 5.2, el umbral que corresponde al punto donde se cruzan las distribuciones de ambas clases, se conoce como punto de igual error (*Equal Error Rate - EER*), y es considerado en muchos casos umbral óptimo. Sin embargo, este umbral puede no ser el mejor debido a la dispersion de las funciones, es decir que en algunos casos puede encontrarse un umbral donde el área de error sea menor que el area de error proporcionada por el *EER*. El punto en el cual el area de error es mínima, es llamado punto de mínimo coste (*Minimum Cost Point - MCP*). Según la teoría de decisión Bayesiana, éste puede ser calculado considerando que el coste en que se incurre es diferente para los dos posibles errores (*falsa aceptación* y *falso rechazo*) [3]. La figura 5.2 muestra de manera gráfica el problema de encontrar el umbral óptimo de decisión. Al escoger un umbral, los registros con puntuaciones mayores o iguales al umbral escogido, son asignadas a la clase 1 (por convención la clase patológica) y las muestras con puntuaciones menores serán asignadas a la clase 2 (normal).

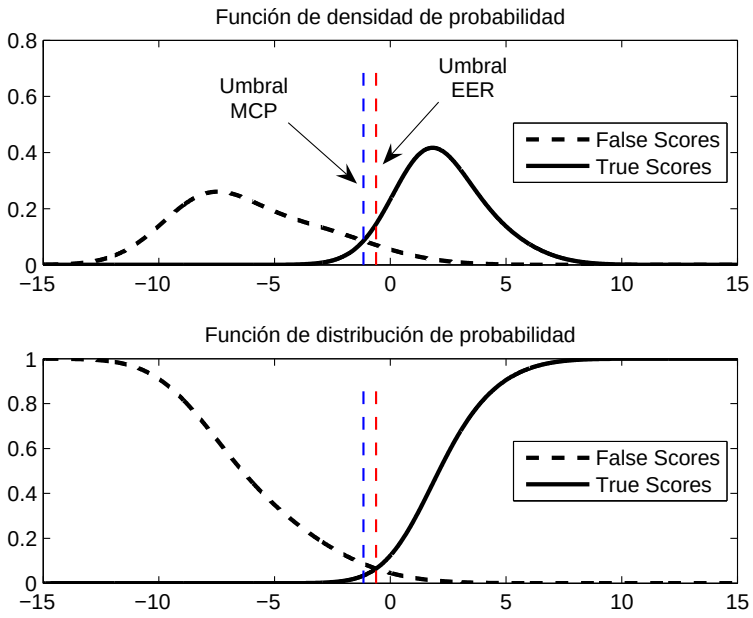


Figura 5.2: Sup. Función de densidad de probabilidad de las puntuaciones de clase normal (*False scores*) y de clase patológica (*True Scores*). Inf. Función de distribución de probabilidad de las puntuaciones de ambas clases. Las líneas punteadas corresponden a dos posibles umbrales de decisión: *Minimum Cost Point - MCP* y *Equal Error Rate - EER*.

Para comparar los resultados obtenidos con los diferentes esquemas de clasificación, los valores de las puntuaciones o umbrales de verosimilitud obtenidos, serán normalizados de tal manera que estén contenidos en el intervalo $[0, 1]$ y puedan ser considerados una

estimación de probabilidad a posteriori [59]. La transformación empleada para tal fin es una función sigmoidal de la forma:

$$f(x) = \frac{1}{1 + e^{-(w_0 + wx)}} \in [0, 1] \tag{5.1}$$

donde,

$$w_0 = \frac{\mu_1^2 - \mu_0^2}{2\sigma^2} \qquad w = \frac{\mu_0 - \mu_1}{\sigma^2} \tag{5.2}$$

donde μ_0 y μ_1 son las medias de las puntuaciones para la clase 0 y para la clase 1 respectivamente. Esta transformación requiere asumir que los valores de las puntuaciones siguen una distribución gaussiana, con varianza común σ^2 [60]. Debido a que en la práctica la dispersion no tiene porqué ser igual, se puede usar el valor de $\sigma = 0{,}5\,(\sigma_0 + \sigma_1)$, donde σ_0 y σ_1 son las estimaciones de las correspondientes desviaciones típicas individuales. Para el caso en el cual las muestras están desbalanceadas un mejor estimador sería de la forma:

$$\sigma = 0,5\left(\frac{n_0}{n}\sigma_0 + \frac{n_1}{n}\sigma_1\right) \tag{5.3}$$

donde n_0 y n_1 son el numero de muestras en la clase 0 y el número de muestras en la clase 1 respectivamente, y n es el número total de muestras.

5.5. Metodología de validación

Para la evaluación de los sistemas se empleará la metodología propuesta en [61], la cual establece como primera medida el empleo de una base de datos disponible para cualquier investigador, y el empleo dentro de ésta de registros que tengan diagnóstico.
Para determinar las capacidades de generalización de los sistemas se adoptará un esquema de validación cruzada, con diferentes conjuntos de entrenamiento-validación (*k-fold*), aleatoriamente escogidos del conjunto completo de datos. En este trabajo se emplean 11-conjuntos, utilizando para el entrenamiento el 70 % de los ficheros y para la validación el 30 % restante. Los resultados finales serán presentados por medio de matrices de confusión (Ver Fig. 5.3). Para construir la matriz de confusión, para el caso en el cual se tienen dos clase (0 y 1), se deben calcular los siguientes parámetros [61]:

- *Detección correcta o aceptación verdadera* (TP, *true positive*): el número (o porcentaje) de patrones de clase 0 que el clasificador asigna correctamente como pertenecientes a la clase 0. Esta medida es llamada también *sensibilidad*
- *Falso rechazo* (FN, *false negative*): el número (o porcentaje) de patrones de clase 0 que el clasificador asigna incorrectamente como pertenecientes a la clase 1.
- *Falsa aceptación* (FP, *false positive*):el número (o porcentaje) de patrones de clase 1 que el clasificador asigna incorrectamente como pertenecientes a la clase 0.
- *Rechazo correcto o verdadero* (TN, *true negative*):el número (o porcentaje) de patrones de clase 1 que el clasificador asigna correctamente como pertenecientes a la clase 1. Esta medida es llamada también *especificidad*.

		Clase real	
		Clase 0	Clase 1
Clase estimada por el clasificador	Clase 0	TP	FP
	Clase 1	FN	TN

Figura 5.3: Aspecto general de una matriz de confusión o contingencia con dos clases.

Nótese que cuando los valores se representan en porcentaje, $TP+FN = 100$ y $FP+TN = 100$. A partir de los valores de las puntuaciones entregadas por cada clasificador, podrán ser construidas las curvas de evaluación de desempeño DET (*Detection Error Trade-off*) y ROC (Característica de Operación del Receptor), y las bandas de confianza en las misma, estimando la desviación estándar en los resultados de los diferentes folds [62]. La curva ROC es una herramienta popular en tareas de decisión médicas [63], expresa el rendimiento en términos de la *sensibilidad* y 1-*especificidad*.
La Curva DET [64] ha sido usada ampliamente en la valoración del desempeño en sistema de identificación de hablante. La curva DET gráfica las tasas de error en ambos ejes (FP y FN), dando tratamiento uniforme a ambos tipos de error. Una explicación más amplia de la metodología de validación, que incluye definiciones y especificaciones de las curvas ROC y DET puede ser encontrada en el apéndice B.

Capítulo 6

Resultados

Los resultados de la metodología propuesta son comparados con los esquemas convencionales de extracción de características y entrenamiento de HMMs. Fueron probadas diferentes configuraciones. 1) En la primera configuración, no se emplea ninguna técnica de extracción de características y se utiliza un clasificador HMM entrenado con el criterio *ML*. 2) En esta configuración, el sistema de detección emplea PCA como técnica de extracción de características y se utiliza un clasificador HMM entrenado con el criterio *ML* (ML_PCA). 3) Esta configuración emplea LDA como técnica de extracción de características y utiliza el mismo criterio para ajustar los parámetros de los HMMs, empleado en las configuraciones anteriores (ML_LDA). 4) En está configuración, no se emplean ninguna técnica de extracción de características, pero los HMMs son entrenados empleando el criterio *MCE*. 5) En este caso, el sistema de detección emplea PCA como técnica de extracción de características y entrena los HMMs a partir del criterio *MCE* (MCE_PCA). 6) Esta configuración es similar a la anterior pero reemplazando la técnica de extracción de características por LDA (MCE_LDA). 7) Esta configuración emplea el método propuesto de extracción de características y entrenamiento simultaneo de los HMMs empleando el criterio *MCE*. En esta configuración se entrena una única transformación $\mathcal{W}$ para los modelos de ambas clases (ver sección 4.1) (MCE_ECD1). 8) Esta configuración al igual que la anterior emplea el método propuesto de extracción de características y entrenamiento simultaneo de los HMMs empleando el criterio *MCE*; sin embargo, en este caso se entrenan dos transformaciones, una para cada clase (modelo) (MCE_ECD2).
Fueron entrenados HMMs con diferentes número de estados (variando en el intervalo $[1, 5]$) y diferente número de mezclas Gaussianas (variando en el intervalo $[2, 5]$), con el objetivo de encontrar el conjunto de parámetros que mejor desempeño presente para la tarea de identificación.

6.1. Resultados sobre la base de datos BD1

La selección de características se realizó empleando la metodología descrita en el apéndice C. Se calcularon los pesos de las variables dinámicas (C.10), que son una representación de la cantidad de información dinámica de cada variable y permiten identifican las variables dinámicas de más influencia en el proceso [58]. La Figura 6.1 muestra los pesos de cada

una de los parámetros considerados en el vector de características (ver sección 5.2). En la

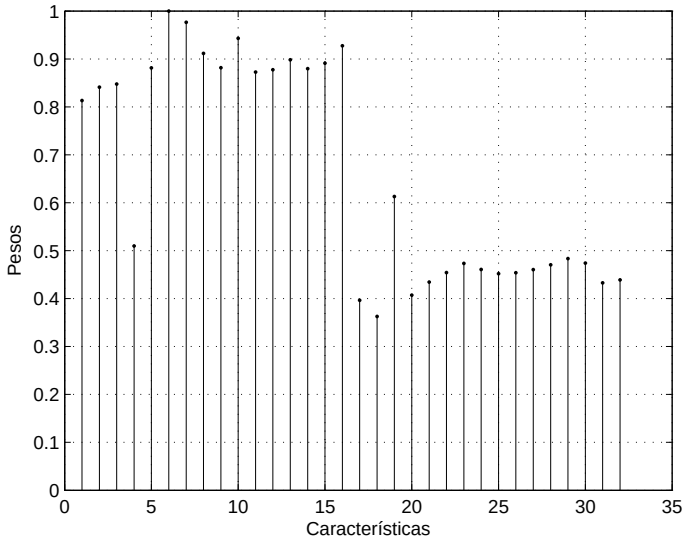


Figura 6.1: Pesos asignados a cada características en la base de datos BD1.

figura 6.1 se puede observar que el conjunto de variables que corresponden a las derivadas, tienen menor peso en relación con las variables originales, por tal motivo son las que menor información aportan al proceso de clasificación. En [58], se determino de que la inclusión de las variables de menor peso en la etapa de clasificación, no mejora el desempeño del sistema. En este trabajo se consideró que las variables que tenían peso mayor al 50 % del máximo, eran las aportaban mayor información al proceso de clasificación. Por esta razón, en lo sucesivo para la base de datos BD1, no serán consideradas las derivadas en el proceso de entrenamiento de los clasificadores, con lo que se busca reducir el costo computacional requerido para el entrenamiento de los HMMs.

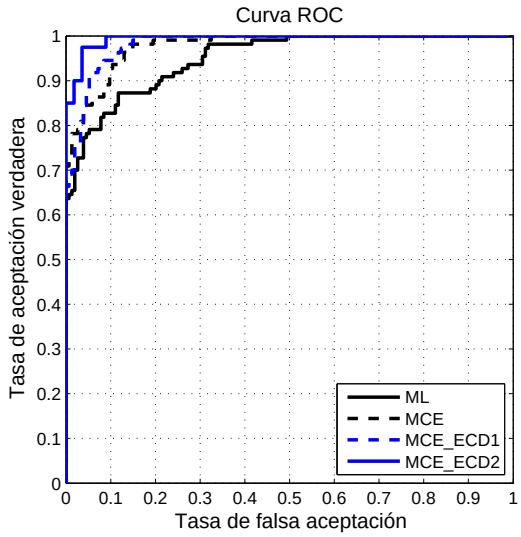
La Tabla 6.1 muestra los mejores resultados obtenidos empleando las diferentes configuraciones descritas anteriormente. Para los casos en los cuales se emplean técnicas de extracción de características, el número entre paréntesis indica la dimensión del espacio para el cual se logró el resultado. En la Tabla 6.1 se puede observar que el empleo del criterio de entrenamiento MCE aplicado a HMMs, reduce en un 6,82 % el error de detección en comparación con el criterio convencional ML. De la misma forma, es posible observar, que el empleo de las técnicas de extracción de características PCA y LDA, no tiene una influencia positiva en el rendimiento. Únicamente para el caso en el cual se entrenó un HMM por el criterio ML, se mejoró el rendimiento al emplear PCA. Sin embargo, la reducción en el error de clasificación está aún por debajo, de la reducción lograda por el criterio de entrenamiento MCE.

Por otro lado, si se compara la reducción del error lograda por la metodología de extracción de características y entrenamiento simultáneo propuesta, se puede observar que la disminución en el error es de 10,32 % para el método que utiliza una transformación para cada modelo y de 9,49 % para el que utiliza únicamente una transformación común a los modelos de ambas clases. La tasa de acierto alcanzada del 94,79 %, reduce en un 11,19 %

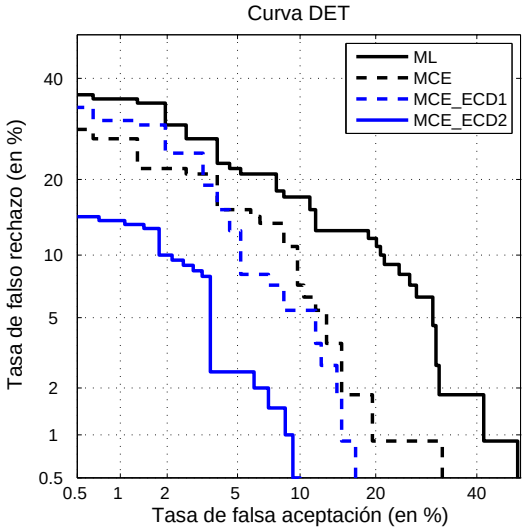
Tabla 6.1: Mejores resultados obtenidos empleando diferentes criterios de entrenamiento de HMM y diferentes técnicas de extracción de características, sobre la base de datos BD1

Configuración	**N° de Estados**	**N° de Gaussianas**	**Matriz de Confusión**	
ML	3	2	82,47 %	12,73 %
			17,53 %	87,27 %
			Eficiencia 84,47 % ± 5,4	
ML_PCA (14)	2	3	88,96 %	9,09 %
			11,04 %	90,91 %
			Eficiencia 86,77 % ± 5,2	
ML_LDA (15)	4	3	79,87 %	8,19 %
			20,13 %	91,81 %
			Eficiencia 84,84 % ± 8,3	
MCE	3	4	90,26 %	7,27 %
			9,74 %	92,73 %
			Eficiencia 91,29 % ± 3,9	
MCE_PCA (14)	4	5	93,88 %	14,29 %
			6,12 %	85,71 %
			Eficiencia 90,48 % ± 4,6	
MCE_LDA (16)	4	4	91,07 %	10,00 %
			8,93 %	90,00 %
			Eficiencia 90,63 % ± 4,5	
MCE_ECD1 (15)	3	5	92,67 %	5,00 %
			7,33 %	95,00 %
			Eficiencia 93,96 % ± 4,3	
MCE_ECD2 (13)	4	3	96,43 %	7,5 %
			3,57 %	92,50 %
			Eficiencia 94,79 % ± 3,2	

el error de clasificación obtenido por otros métodos sobre esta misma base de datos [58]. De igual manera, es posible observar que las desviación estándar de la eficiencia en todos los casos en los cuales se empleo el criterio MCE, para ajustar los parámetros de los modelos, es menor que la desviación para el caso en el cual se utilizó el criterio ML.
Las Figuras 6.2(a) y 6.2(b) muestran de manera comparativa las curvas ROC y DET, para el sistema empleando el criterio convencional ML, el criterio MCE y la metodología propuesta. El área bajo la curva ROC (AUC) es una medida escalar que en algunos trabajos se ha empleado como un buen predictor de la eficiencia del sistema [65]. La Tabla 6.2 muestra el área bajo la curva para las mismas configuraciones comparadas en la Figura 6.2. Se puede observar que la mayor AUC es para la configuración MCE_ECD2. Las Figura 6.3(a) y 6.3(b) muestran de manera comparativa las curvas ROC y DET, para el sistema empleando el criterio de entrenamiento MCE y diferentes métodos de reducción empleados. La Tabla 6.3 muestra las AUC para las configuraciones de la figura 6.3. Como se puede observar el AUC cuando se empleó PCA y LDA fue menor que para el caso



(a) Curvas ROC para el sistema empleando diferentes criterios de entrenamiento de HMMs sobre la base de datos BD1.

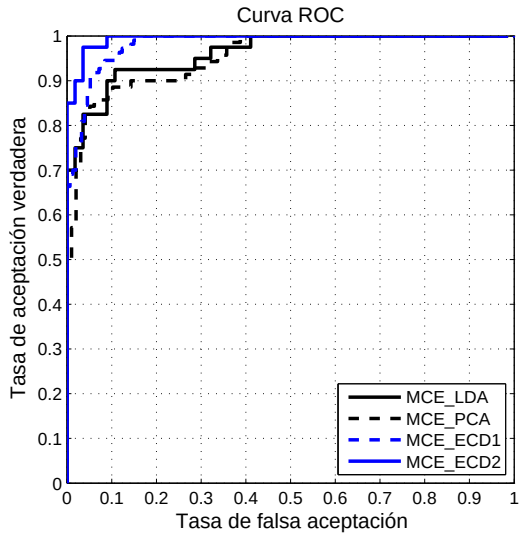


(b) Curvas DET para el sistema empleando diferentes criterios de entrenamiento de HMMs sobre la base de datos BD1.

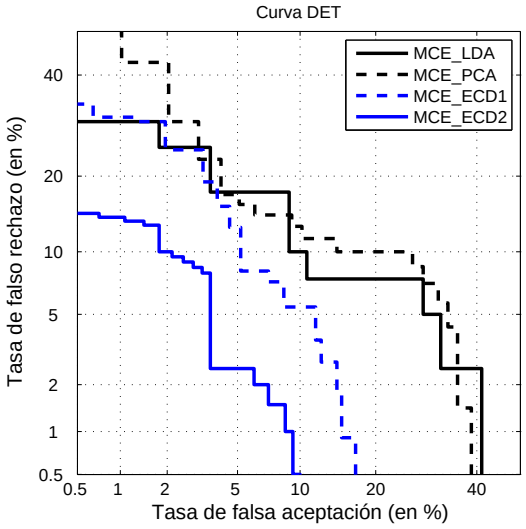
Figura 6.2: Curvas DET y ROC para el sistema empleando diferentes criterios de entrenamiento sobre la base de datos BD1.

Tabla 6.2: Área bajo las curvas ROC para el sistema empleando diferentes criterios de entrenamiento de HMMs obre la base de datos BD1.

Configuración	**AUC**
ML	0.9507
MCE	0.9780
MCE_ECD1	0.9822
MCE_ECD2	0.9942



(a) Curvas ROC para el sistema empleando el criterio de entrenamiento MCE y diferentes métodos de reducción sobre la base de datos BD1.



(b) Curvas DET para el sistema empleando el criterio de entrenamiento MCE y diferentes métodos de reducción sobre la base de datos BD1.

Figura 6.3: Curvas DET y ROC para el sistema empleando el criterio de entrenamiento MCE y diferentes métodos de reducción sobre la base de datos BD1.

cuando no se empleó ninguna técnica de reducción. Este hecho muestra que el desempeño esperado del sistema se reduce cuando se emplean los métodos convencionales de extracción de características, contrario a lo sucedido cuando se emplea el método propuesto. Adicionalmente, la curva DET para la configuración MCE_DFE2 está mucho más cerca a la esquina inferior izquierda del plano (punto ideal) en comparación con las curvas DET de los demás métodos evaluados.

Uno de los planteamientos sobre los cuales se basa este trabajo, es el hecho de que emplear

Tabla 6.3: Área bajo las curvas ROC para el sistema empleando el criterio de entrenamiento MCE y diferentes métodos de reducción sobre la base de datos BD1.

Configuración	AUC
MCE_LDA	0.9616
MCE_PCA	0.9528
MCE_ECD1	0.9822
MCE_ECD2	0.9942

en el entrenamiento una medida de distancia entre los modelos de clases diferentes, permite obtener modelos más disímiles entre las clases. La Figura 6.4, muestra diferentes medidas distancia entre los modelos a través de las iteraciones, para una ejecución del algoritmo; además se muestra la evolución de la función de pérdida Ec. (1.33). De la Figura 6.4

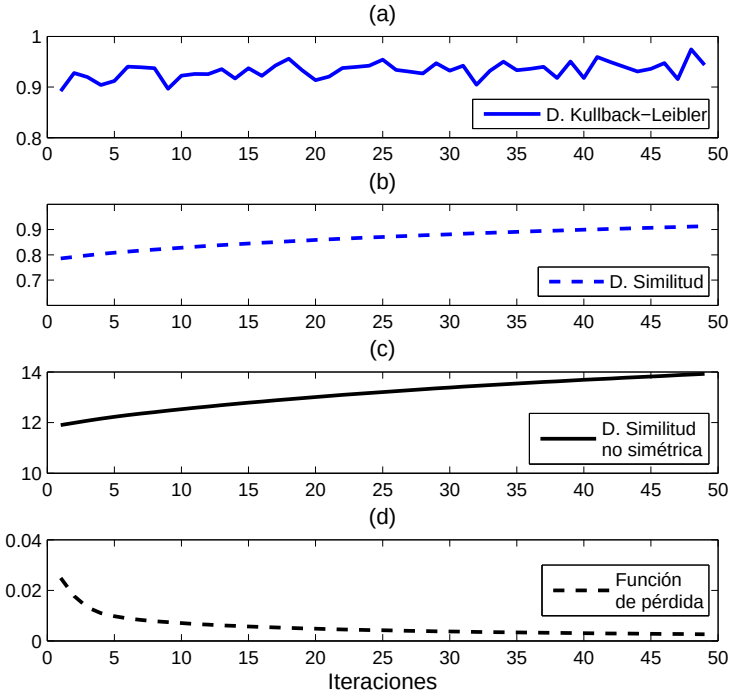


Figura 6.4: Medidas de distancia entre los modelos HMMs a través de las iteraciones del algoritmo MCE para la base de datos BD1. (a) Distancia de Kullback-Leibler. (b) Distancia a partir de la medida de Similitud. (c) Distancia no simétrica a partir de la medida de Similitud. (d) Función de pérdida.

es claro que la medida de distancia basada en la divergencia de *Kullback-Leibler*, no es consistente a lo largo de la ejecución del algoritmo, teniendo en cuenta que la función de pérdida disminuye durante toda la ejecución. Es de notar, que después de la 6 iteración la tasa de entrenamiento alcanzo el 100 %; sin embargo, el algoritmo continuó aumenta la distancia entre los modelos, como lo muestran las dos medidas basadas en la distancia por Similitud descritas en el capítulo 2, las cuales aumentan conforme la función de pérdida disminuye.

6.2. Resultados sobre la base de datos BD2

De igual forma a como se realizó la selección de características sobre la base de datos BD1, fue realizada sobre la base de datos BD2. La Figura 6.5 muestra los pesos de cada una de los parámetros considerados en el vector de características. De la Figura 6.5, se puede ob-

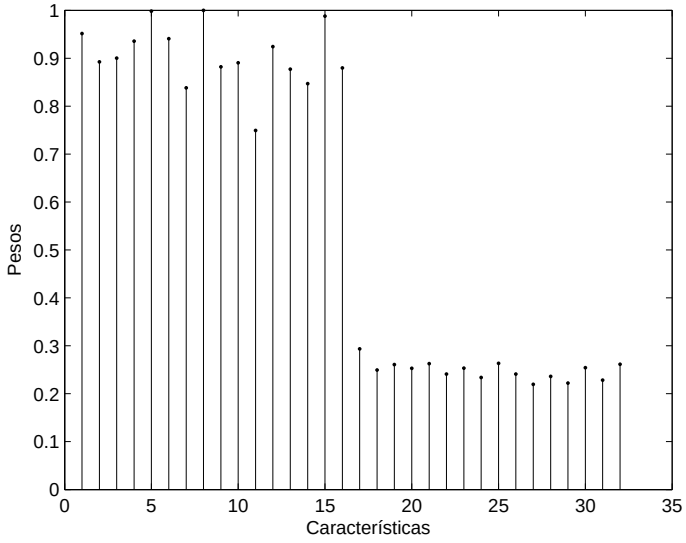


Figura 6.5: Pesos asignados a cada características en la base de datos BD2.

servar que para la base de datos BD2, los pesos de las derivadas de las variables originales, son aún menores con relación a las variables originales, en comparación con lo obtenido para la base de datos BD1. Por consiguiente, en las pruebas realizadas sobre la base de datos BD2, no fueron consideradas las derivadas dentro del vector de características.
La Tabla 6.4 muestra los mejores resultados obtenidos empleando las configuraciones descritas al comienzo de este capítulo. Se puede observar que en este caso, la reducción del error empleando el criterio MCE en comparación con el criterio ML, no es muy significativa. Sin embargo, el método propuesto, nuevamente mejora el desempeño del sistema. El espacio de entrenamiento en este caso pudo ser reducido a 10, mostrando así una reducción del 37,5 % con respecto a la dimensión del espacio original. De igual forma que para la base de datos BD1, se puede observar a partir de la Tabla 6.4, que las desviaciones estándar de la eficiencia, en la mayoría de los casos en los cuales se empleo el criterio MCE para ajustar los parámetros de los modelos, son menores que las desviaciones para el caso en el cual se utilizó el criterio ML.
Las Figuras 6.6(a) y 6.6(b) muestran de manera comparativa las curvas ROC y DET, para el sistema empleando el criterio convencional ML, el criterio MCE y la metodología propuesta. La Tabla 6.5 muestra las AUC para las configuraciones comparadas en la Figura 6.6. Se puede observar que la mayor AUC es nuevamente para la configuración MCE_ECD2. Sin embargo, en la Figura 6.6 no se observa una diferencia notable entre las cuatro configuraciones. Únicamente en la Figura 6.6(b), la curva DET correspondiente a la configuración MCE_ECD2, se muestra un poco más cercana al vértice inferior izquierdo.

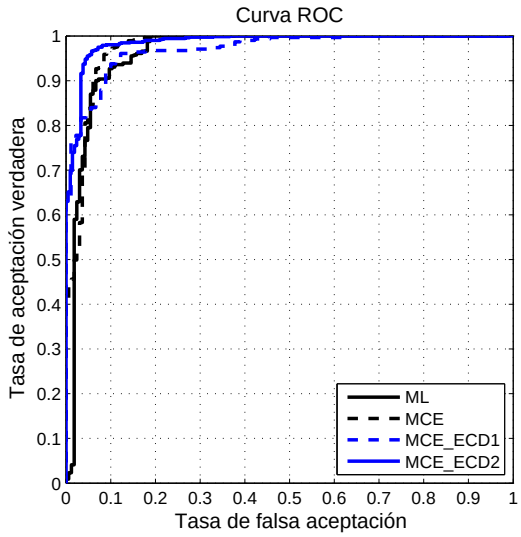
Tabla 6.4: Mejores resultados obtenidos empleando diferentes criterios de entrenamiento de HMM y diferentes técnicas de extracción de características, sobre la base de datos BD2

Configuración	**N° de Estados**	**N° de Gaussianas**	**Matriz de Confusión**	
ML	2	2	93,58 %	12,12 %
			6,42 %	87,88 %
			Eficiencia 92,27 % ± 3,1	
ML_PCA (15)	3	5	94,11 %	10,91 %
			5,89 %	89,09 %
			Eficiencia 92,98 % ± 6,1	
ML_LDA (13)	4	4	94,47 %	15,76 %
			5,53 %	84,24 %
			Eficiencia 92,15 % ± 6,2	
MCE	2	4	95,19 %	8,48 %
			4,81 %	91,52 %
			Eficiencia 94,35 % ± 2,36	
MCE_PCA (10)	5	3	96,81 %	26,67 %
			3,19 %	73,33 %
			Eficiencia 91,48 % ± 3,1	
MCE_LDA (12)	4	4	96,51 %	21,48 %
			3,49 %	78,52 %
			Eficiencia 92,42 % ± 2,1	
MCE_ECD1 (10)	2	4	96,08 %	12,22 %
			3,92 %	87,78 %
			Eficiencia 94,19 % ± 2,23	
MCE_ECD2 (10)	2	4	98,04 %	10,00 %
			1,96 %	90,00 %
			Eficiencia 96,21 % ± 1,07	

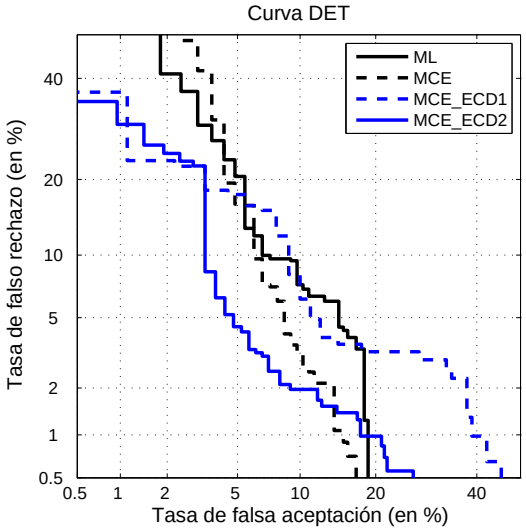
Tabla 6.5: Área bajo las curvas ROC para el sistema empleando diferentes criterios de entrenamiento de HMMs obre la base de datos BD2.

Configuración	**AUC**
ML	0.9507
MCE	0.9725
MCE_ECD1	0.9706
MCE_ECD2	0.9846

Las Figura 6.7(a) y 6.7(b) muestran de manera comparativa las curvas ROC y DET, para el sistema empleando el criterio de entrenamiento por MCE y los diferentes métodos de reducción empleados. La Tabla 6.6 muestra las AUC para las configuraciones de la figura 6.7. Como se puede observar el AUC cuando se empleó PCA y LDA fue mucho menor que para el caso cuando no se empleó ninguna técnica de reducción. Este hecho ratifica que el desempeño esperado del sistema se reduce cuando se emplean los métodos convencionales de extracción de características sobre este tipo de características dinámicas. De la Figura



(a) Curvas ROC para el sistema empleando diferentes criterios de entrenamiento de HMMs sobre la base de datos BD2.

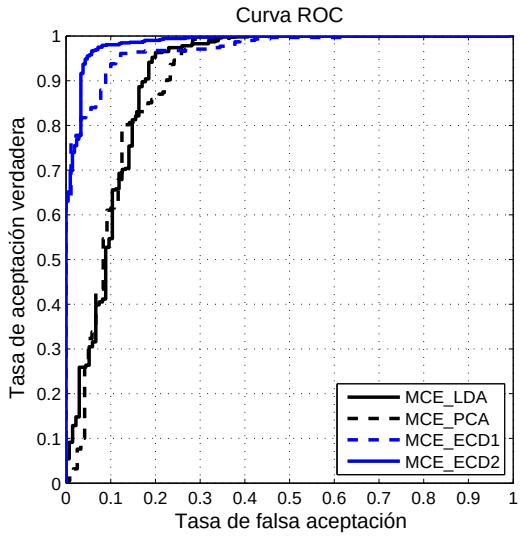


(b) Curvas DET para el sistema empleando diferentes criterios de entrenamiento de HMMs sobre la base de datos BD2.

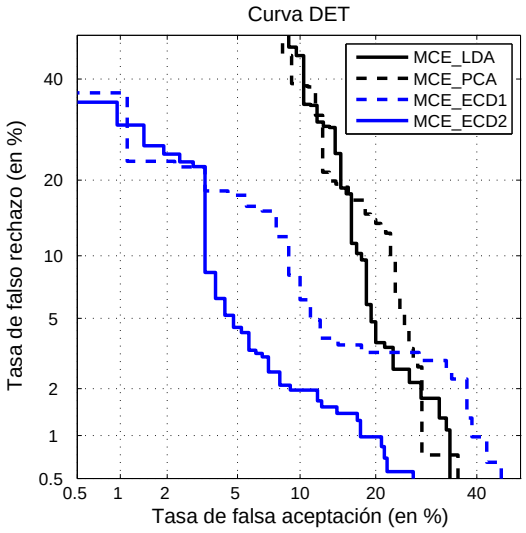
Figura 6.6: Curvas DET y ROC para el sistema empleando diferentes criterios de entrenamiento sobre la base de datos BD2.

Tabla 6.6: Área bajo las curvas ROC para el sistema empleando el criterio de entrenamiento MCE y diferentes métodos de reducción sobre la base de datos BD2.

Configuración	**AUC**
MCE_LDA	0.9040
MCE_PCA	0.8974
MCE_ECD1	0.9706
MCE_ECD2	0.9846



(a) Curvas ROC para el sistema empleando el criterio de entrenamiento MCE y diferentes métodos de reducción sobre la base de datos BD2.



(b) Curvas DET para el sistema empleando el criterio de entrenamiento MCE y diferentes métodos de reducción sobre la base de datos BD1.

Figura 6.7: Curvas DET y ROC para el sistema empleando el criterio de entrenamiento MCE y diferentes métodos de reducción sobre la base de datos BD2.

6.7 se puede observar que existe una notable diferencia en el desempeño del sistema, cuando se emplean los métodos convencionales de extracción de características y cuando se emplea el método propuesto. De igual forma que lo sucedido para la base de datos BD1, se puede observar en la base de datos BD2, que es mejor el rendimiento para la configuración MCE_ECD2 que para la configuración MCE_ECD1.

Aunque no se observa un incremento considerable en el desempeño del sistema, cuando se emplea la metodología propuesta sobre la base de datos BD2, se puede resaltar, que

la utilización de ésta no genera efectos negativos en el rendimiento y permite reducir el espacio de entrenamiento de los HMMs.
La Figura 6.8, muestra diferentes medidas distancia entre los modelos a través de las iteraciones, para una ejecución del algoritmo; además se muestra la evolución de la función de pérdida para la base de datos BD2. Se puede observar de la Figura 6.4 que en este caso,

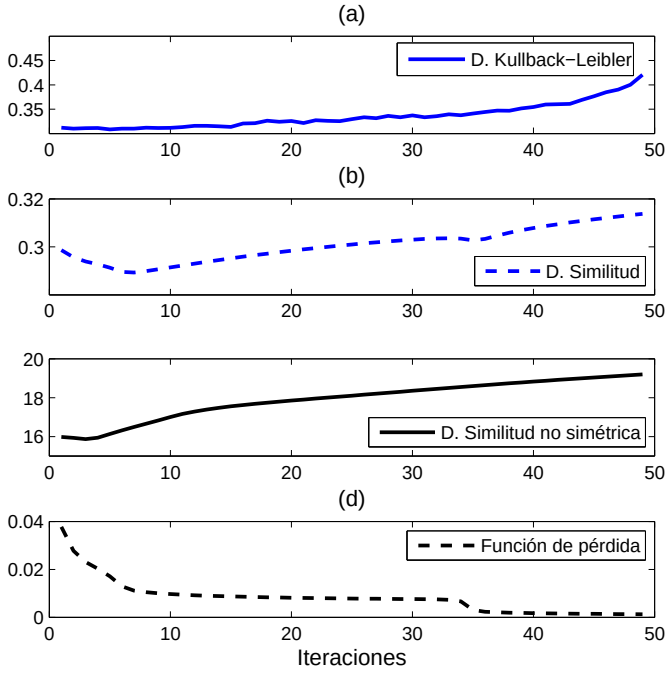


Figura 6.8: Medidas de distancia entre los modelos HMMs a través de las iteraciones del algoritmo MCE para la base de datos BD2. (a) Distancia de Kullback-Leibler. (b) Distancia a partir de la medida de Similitud. (c) Distancia no simétrica a partir de la medida de Similitud. (d) Función de pérdida.

la distancia derivada de la divergencia de *Kullback-Leibler*, se comporta mejor que para el caso de la base de datos BD1. Sin embargo, presenta algunas oscilaciones.
Un efecto negativo del método propuesto, es el aumento en el coste computacional, que se genera al realizar simultáneamente la estimación de la transformación y de los parámetros del HMM. El incremento en el tiempo computacional, varía con respecto al número de características y al número de observaciones con el que se cuenta para el entrenamiento. En el caso de la base de datos BD2 que cuenta con un número mayor de sujetos que la base de datos BD1, se estimó que el tiempo de cálculo de cada iteración para el método propuesto, aumento aproximadamente 10 veces, con relación al tiempo de una iteraciones del algoritmo MCE convencional. Este hecho aunque implica mayor tiempo en la etapa de entrenamiento, no se ve reflejado en la ejecución de un proceso en línea, debido a que hace parte del los procesos de entrenamiento y ajuste de los sistemas, que se realizan fuera de línea.

Capítulo 7

Discusión y Conclusiones

Los resultados presentados para la base de datos BD1, muestran un claro aumento en el desempeño del sistema cuando se ajustan los parámetros de los HMMs por medio del criterio MCE. Además, el ajuste simultáneo de la etapa de extracción de características y del modelo HMM, disminuyo aún más el error de clasificación con respecto al método convencional de entrenamiento por medio del criterio ML. Es claro que la configuración denominada MCE_ECD2, es decir, la que emplea una transformación independiente para cada modelo, presento mejor desempeño que la configuración MCE_ECD1. Este hecho puede ser explicado, debido a que cuando se utiliza una única transformación para los modelos de ambas clases, se relaciona el evento asociado al estado i en el modelo 1, con el evento asociado al estado i en el modelo 2. Estos dos eventos no tienen porque estar relacionados, si tenemos en cuenta que cada modelo pertenece a un proceso diferente y que la secuencia de estados para una secuencia de observación dada, puede ser diferente en cada modelo. La configuración MCE_ECD2 entrega mejores resultados al tratar cada modelo de manera independiente.

Con relación a la base de datos BD2, el aumento en el desempeño debido a la utilización de algoritmos derivados del criterio MCE, no fue tan notorio como para el caso de la base de datos BD1. Sin embargo, se obtuvo una reducción del espacio de entrenamiento del 37,5 %, sin degradar el rendimiento del sistema. Adicionalmente, si se compara el rendimiento de los sistemas en los cuales se empleó la metodología propuesta, con respecto a los sistemas en los cuales se emplearon como métodos de extracción de características PCA y LDA, se puede observar una gran diferencia a favor de la metodología propuesta. Este hecho se ve claramente a partir de las curvas ROC y DET (Figura 6.6).
La metodología propuesta entrego los mejores resultados en cuanto a disminución del error de clasificación, para el problema de detección abordado en este trabajo. Además, la metodología permitió encontrar un espacio de entrenamiento de menor dimensión. Aunque la disminución del espacio de entrenamiento para la base de datos BD1, no fue mayor a 3, es de notar que con la transformación se disminuyó aún más el error de clasificación, lo cual no sucede cuando se emplean los métodos convencionales de extracción de características. Además, debido a que la metodología está basada en el algoritmo MCE, permite seguir maximizando la distancia entre los modelos de clases diferentes, aún cuando la tasa de entrenamiento ya esté en el máximo.

Por otro lado, es claro que la metodología propuesta aumenta considerablemente el costo computacional; sin embargo, se debe tener en cuenta que el proceso para el cual es utilizada, se realiza fuera de línea.
Una de las razones fundamentales por las cuales es evidente la disminución del error, a través del empleo del algoritmo MCE, es debido a que éste basa directamente el ajuste de los parámetros, sobre una función de costo que evalúa el error de clasificación en cada iteración, a partir de una medida de distancia entre los modelos. De está manera, los parámetro de cada modelo son ajustados teniendo en cuenta, no sólo la información de la clase a la que pertenecen, sino también de la clase contraria. Sin embargo, éste hecho no permite utilizar el algoritmo MCE para entrenar un único modelo, útil en algunos casos en los cuales sólo se cuenta con información etiquetada de una sola clase.
Por otro lado, después del análisis realizado en el Capítulo 2 a diferentes medidas de distancia entre HMMs, se concluyó que, de las medidas estudiadas, sólo podía ser empleada en el entrenamiento la distancia no simétrica a partir de la medida de similitud. Lo anterior se debe a las siguientes razones:

- La distancia no simétrica a partir de la medida de similitud, puede ser generalizada para su aplicación a modelos continuos.
- El cálculo de la media de distancia *Kullback-Leibler*, se realiza generando un nuevo conjunto de secuencias de observación a partir de los modelos entrenados, con lo cual, no se emplea la información del conjunto de datos de entrenamiento para medir la distancia entre los modelos. Por el contrario, el cálculo de la distancia no simétrica a partir de la medida de similitud, se realiza empleando la información de las secuencias de observaciones, utilizadas para el ajuste de los parámetros del HMM.
- El empleo de la medida de distancia por similitud (simétrica), se dificulta debido al hecho de que en cada instante del algoritmo se emplea sólo una secuencia de observación para ajustar los parámetros de alguno de los modelos; mientras que la medida simétrica requiere el empleo de dos secuencias de observación (una de cada clase).

La metodología de validación empleada, permite obtener resultados más claros y proporciona mayores herramientas para obtener una valoración objetiva del desempeño de los sistemas. El empleo de curvas de rendimiento da una mayor idea del comportamiento real del sistema y de qué tan amplio es el rango en el cual se puede desplazar el umbral de decisión, sin afectar de manera significativa el desempeño del sistema.
Aunque en el trabajo presentado se evaluó la influencia del número de estados y del número de mezclas Gaussianas del HMM, en el rendimiento del sistema, existen una serie de parámetros asociados al algoritmo de entrenamiento, que requieren de un estudio detallado acerca del efecto de éstos, de tal forma que puede ser encontrado un conjunto de parámetros óptimo, que permita aumentar la tasa de convergencia y disminuir el tiempo computacional del algoritmo. Una posible forma de abordar el problema, es mediante algoritmos genéticos debido a que el espacio de búsqueda es amplio.
Adicionalmente, se debe considerar que el algoritmo GPD en el cual está basado el criterio de entrenamiento MCE, no garantiza la convergencia a un mínimo global de la función de

pérdida. Una investigación en está línea, podría generar abordar el problema de construir la metodología de reducción de espacios, sobre un algoritmo más robusto, es decir, de mejores propiedades de convergencia.

Parte IV

Apéndices

Apéndice A

Reestimación de los parámetros de HMMs por medio de MCE

Para utilizar el algoritmo GPD generalizado (1.34) en la estimación de parámetros de un modelo oculto de Markov, se deben definir las siguientes transformaciones de parámetros, que permiten mantener las restricciones probabilísticas de los HMM durante la adaptación:

$$\mathbf{\Pi}_{jk} \to \tilde{\mathbf{\Pi}}_{jk} \quad \text{donde} \quad \mathbf{\Pi}_{jk} = \frac{\exp\left(\tilde{\mathbf{\Pi}}_{jk}\right)}{\sum_{l=1}^{n_\vartheta} \exp\left(\tilde{\mathbf{\Pi}}_{jl}\right)} \tag{A.1}$$

$$\boldsymbol{p}_{\theta_1}(j) \to \tilde{\boldsymbol{p}}_{\theta_1}(j) \quad \text{donde} \quad \boldsymbol{p}_{\theta_1}(j) = \frac{\exp\left(\tilde{\boldsymbol{p}}_{\theta_1}(j)\right)}{\sum_{l=1}^{n_\vartheta} \exp\left(\tilde{\boldsymbol{p}}_{\theta_1}(l)\right)} \tag{A.2}$$

Para la adaptación de parámetros de las componentes Gaussianas del modelo, se asume por simplicidad que la matrix de covarianza $\Sigma_{jr} = [\sigma_{jrp}^2]_{p=1}^{\rho}$ se asume diagonal.

$$c_{jr} \to \tilde{c}_{jr} \quad \text{donde} \quad c_{jr} = \frac{\exp(\tilde{c}_{jr})}{\sum_{l=1}^{M} \exp(\tilde{c}_{jl})} \tag{A.3}$$

$$\mu_{jrp} \to \tilde{\mu}_{jrp} \quad \text{donde} \quad \tilde{\mu}_{jrp} = \frac{\mu_{jrp}}{\sigma_{jrp}} \tag{A.4}$$

$$\sigma_{jrp} \to \tilde{\sigma}_{jrp} \quad \text{donde} \quad \tilde{\sigma}_{jrp} = \log \sigma_{jrp} \tag{A.5}$$

A.1. Reestimación del vector probabilidad de estado inicial

Partiendo de la definición para la actualización de parámetros dada por el algoritmo GPD (1.34), la actualización del vector probabilidad inicial está dad por:

$$\tilde{p}_{\theta_1 j}^{(i)}(n+1) = \tilde{p}_{\theta_1 j}^{(i)}(n) - \varepsilon \left. \frac{\partial \ell_i(\varphi_n; \lambda)}{\partial \tilde{p}_{\theta_1 j}^{(i)}} \right|_{\lambda=\lambda_n} \tag{A.6}$$

Por regla de la cadena,

$$\frac{\partial \ell_i(\varphi_n;\lambda)}{\partial \tilde{p}^{(i)}_{\theta_1 j}} = \frac{\partial \ell_i(d_i)}{\partial d_i}\frac{\partial d_i}{\partial \tilde{p}^{(i)}_{\theta_1 j}} \tag{A.7}$$

El desarrollo del primer factor de la ecuación (A.7), teniendo en cuenta la definición (1.32), aplica para la actualización de todos los parámetros derivados de la minimización de (1.33). Tenemos,

$$\frac{\partial \ell_i(d_i)}{\partial d_i} = \frac{\gamma \exp(-\gamma d_i + \alpha)}{(1+\exp(-\gamma d_i+\alpha))^2} = \gamma \ell_i^2(d_i)\exp(-\gamma d_i+\alpha) \tag{A.8}$$

despejando de (1.32),

$$\exp(-\gamma d_i+\alpha) = \frac{1-\ell_i(d_i)}{\ell_i(d_i)} \tag{A.9}$$

y reemplazando (A.9) en (A.8), se tiene:

$$\frac{\partial \ell_i(d_i)}{\partial d_i} = \gamma \ell_i(d_i)(1-\ell_i(d_i)) \tag{A.10}$$

Ahora, para completar la función de actualización del vector probabilidad inicial, es necesario desarrollar el segundo término de la ecuación (A.7). Teniendo en cuenta las ecuaciones (1.29) y (1.31), se tiene,

$$\frac{\partial d_i}{\partial \tilde{p}^{(i)}_{\theta_1 j}} = \frac{\partial}{\partial \tilde{p}^{(i)}_{\theta_1 j}} \log p^{(i)}_{\theta_1 \bar{\theta}_0} = \delta(\bar{\theta}_0 - j)\frac{\partial}{\partial \tilde{p}^{(i)}_{\theta_1 j}}\left(\log p^{(i)}_{\theta_1 j}\right) \tag{A.11}$$

$$\frac{\partial}{\partial \tilde{p}^{(i)}_{\theta_1 j}} \log p^{(i)}_{\theta_1 j} = \frac{1}{p^{(i)}_{\theta_1 j}}\frac{\partial}{\partial \tilde{p}^{(i)}_{\theta_1 j}}\left(\frac{\exp\left(\tilde{p}^{(i)}_{\theta_1 j}\right)}{\sum_{l=1}^{n_\vartheta}\exp\left(\tilde{p}^{(i)}_{\theta_1 l}\right)}\right) = 1 - p^{(i)}_{\theta_1 j} \tag{A.12}$$

A.2. Reestimación de la matriz probabilidad de transición de estados

Se puede mostrar que para una secuencia $\boldsymbol{\varphi}_n \in C_i$ del conjunto de entrenamiento, el ajuste discriminativo del parámetro $\tilde{\mathbf{\Pi}}$ partiendo de la definición (1.34), esta dado por:

$$\tilde{\mathbf{\Pi}}^{(i)}_{jk}(n+1) = \tilde{\mathbf{\Pi}}^{(i)}_{jk}(n) - \varepsilon \left.\frac{\partial \ell_i(\varphi_n;\lambda)}{\partial \tilde{\mathbf{\Pi}}^{(i)}_{jk}}\right|_{\lambda=\lambda_n} \tag{A.13}$$

Por regla de la cadena,

$$\frac{\partial \ell_i(\varphi_n;\lambda)}{\partial \tilde{\Pi}^{(i)}_{jk}} = \frac{\partial \ell_i(d_i)}{\partial d_i}\frac{\partial d_i}{\partial \tilde{\Pi}^{(i)}_{jk}} \tag{A.14}$$

Teniendo en cuenta el resultado mostrado en (A.10), nos interesa ahora desarrollar el segundo factor de la ecuación (A.14). Tenemos,

$$\frac{\partial d_i}{\partial \tilde{\Pi}^{(i)}_{jk}} = -\frac{\partial g_i(\varphi_n;\lambda)}{\partial \tilde{\Pi}^{(i)}_{jk}} = -\sum_{t=1}^{n_\varphi}\frac{\partial \log \Pi^{(i)}_{\bar{\theta}_{t-1}\bar{\theta}_t}}{\partial \tilde{\Pi}^{(i)}_{jk}} \tag{A.15}$$

$$\frac{\partial d_i}{\partial \tilde{\Pi}_{jk}^{(i)}} = -\sum_{t=1}^{n_\varphi} \delta\left(\bar{\theta}_{t-1} - j\right)\left(\bar{\theta}_t - k\right) \frac{\partial \log \Pi_{jk}^{(i)}}{\partial \tilde{\Pi}_{jk}^{(i)}} \tag{A.16}$$

luego,

$$\frac{\partial \log \Pi_{jk}^{(i)}}{\partial \tilde{\Pi}_{jk}^{(i)}} = \frac{1}{\Pi_{jk}^{(i)}} \frac{\partial}{\partial \tilde{\Pi}_{jk}^{(i)}} \left(\frac{\exp\left(\tilde{\Pi}_{jk}^{(i)}\right)}{\sum_{l=1}^{n_\vartheta} \exp\left(\tilde{\Pi}_{jl}^{(i)}\right)} \right) = \frac{1}{\Pi_{jk}^{(i)}} \left(\Pi_{jk}^{(i)} - \left(\Pi_{jk}^{(i)}\right)^2 \right) \tag{A.17}$$

$$\frac{\partial \log \Pi_{jk}^{(i)}}{\partial \tilde{\Pi}_{jk}^{(i)}} = \left(1 - \Pi_{jk}^{(i)}\right) \tag{A.18}$$

A.3. Reestimación de los parámetros de las mezclas Gaussianas del modelo

A.3.1. Actualización del vector de medias

El ajuste discriminativo del parámetro μ partiendo de la definición (1.34), esta dado por:

$$\tilde{\mu}_{jrm}^{(i)}(n+1) = \tilde{\mu}_{jrm}^{(i)}(n) - \varepsilon \left. \frac{\partial \ell_i(\varphi_n; \lambda)}{\partial \tilde{\mu}_{jrm}^{(i)}} \right|_{\lambda=\lambda_n} \tag{A.19}$$

Por regla de la cadena,

$$\frac{\partial \ell_i(\varphi_n; \lambda)}{\partial \tilde{\mu}_{jrm}^{(i)}} = \frac{\partial \ell_i(d_i)}{\partial d_i} \frac{\partial d_i}{\partial \tilde{\mu}_{jrm}^{(i)}} \tag{A.20}$$

Teniendo en cuenta el resultado mostrado en (A.10), nos interesa ahora desarrollar el segundo factor de la ecuación (A.20). Tenemos,

$$\frac{\partial d_i}{\partial \tilde{\mu}_{jrm}^{(i)}} = -\frac{\partial g_i(\varphi_n; \lambda)}{\partial \tilde{\mu}_{jrm}^{(i)}} = -\sum_{t=1}^{n_\varphi} \frac{\partial \log\left(b_{\bar{\theta}_t}^{(i)}(\varphi_t)\right)}{\partial \tilde{\mu}_{jrm}^{(i)}} \tag{A.21}$$

$$\frac{\partial d_i}{\partial \tilde{\mu}_{jrm}^{(i)}} = -\sum_{t=1}^{n_\varphi} \delta\left(\bar{\theta}_t - j\right) \frac{\partial \log\left(b_j^{(i)}(\varphi_t)\right)}{\partial \tilde{\mu}_{jrm}^{(i)}} \tag{A.22}$$

Ahora,

$$\begin{aligned} \frac{\partial \log\left(b_j^{(i)}(\varphi_t)\right)}{\partial \tilde{\mu}_{jrm}^{(i)}} = \frac{1}{b_j^{(i)}(\varphi_t)} c_{jr}^{(i)} \left|\Sigma_{jr}^{(i)}\right|^{-\rho/2} \left(\frac{\varphi_{tl} - \mu_{jrl}^{(i)}}{\sigma_{jrl}^{(i)}}\right) \ldots \\ \exp\left(-\frac{1}{2} \sum_{l=1}^{\rho} \left(\frac{\varphi_{tl} - \mu_{jrl}^{(i)}}{\sigma_{jrl}^{(i)}}\right)^2\right) \end{aligned} \tag{A.23}$$

Finalmente,

$$\begin{aligned}\tilde{\mu}_{jrm}^{(i)}(n+1) = \tilde{\mu}_{jrm}^{(i)}(n) + \epsilon\gamma\ell\left(d_i\left(\boldsymbol{\varphi}_n\right)\right)\left(1-\ell\left(d_i\left(\boldsymbol{\varphi}_n\right)\right)\right) \\ \frac{1}{n_\varphi}\sum_{t=1}^{n_\varphi}\delta\left(\bar{\theta}_t - j\right)c_{jr}^{(i)}\left(b_j^{(i)}\left(\varphi_t\right)\right)^{-1}\left|\Sigma_{jr}^{(i)}\right|^{-1/2}(2\pi)^{-\rho/2} \\ \exp\left(-\frac{1}{2}\sum_{l=1}^{\rho}\left(\frac{\varphi_{tl}-\mu_{jrl}^{(i)}(n)}{\sigma_{jrl}^{(i)}(n)}\right)^2\right)\left(\frac{\varphi_{tm}}{\sigma_{jrm}^{(i)}(n)} - \tilde{\mu}_{jrm}^{(i)}(n)\right)\end{aligned} \tag{A.24}$$

A.3.2. Actualización de la matriz de covarianza

El ajuste discriminativo del parámetro Σ partiendo de la definición (1.34) y expresado en términos de las componentes de la matriz Σ, esta dado por:

$$\tilde{\sigma}_{jrm}^{(i)}(n+1) = \tilde{\sigma}_{jrm}^{(i)}(n) - \varepsilon\left.\frac{\partial\ell_i\left(\varphi_n;\lambda\right)}{\partial\tilde{\sigma}_{jrm}^{(i)}}\right|_{\lambda=\lambda_n} \tag{A.25}$$

Por regla de la cadena,

$$\frac{\partial\ell_i\left(\varphi_n;\lambda\right)}{\partial\tilde{\sigma}_{jrm}^{(i)}} = \frac{\partial\ell_i\left(d_i\right)}{\partial d_i}\frac{\partial d_i}{\partial\tilde{\sigma}_{jrm}^{(i)}} \tag{A.26}$$

Teniendo en cuenta el resultado mostrado en (A.10), se debe desarrollar el segundo factor de la ecuación (A.26). Se Tiene,

$$\frac{\partial d_i}{\partial\tilde{\sigma}_{jrm}^{(i)}} = -\frac{\partial g_i\left(\varphi_n;\lambda\right)}{\partial\tilde{\sigma}_{jrm}^{(i)}} = -\sum_{t=1}^{n_\varphi}\frac{\partial\log\left(b_{\bar{\theta}_t}^{(i)}\left(\varphi_t\right)\right)}{\partial\tilde{\sigma}_{jrm}^{(i)}} \tag{A.27}$$

$$\frac{\partial d_i}{\partial\tilde{\sigma}_{jrm}^{(i)}} = -\sum_{t=1}^{n_\varphi}\delta\left(\bar{\theta}_t - j\right)\frac{\partial\log\left(b_j^{(i)}\left(\varphi_t\right)\right)}{\partial\tilde{\sigma}_{jrm}^{(i)}} \tag{A.28}$$

Ahora,

$$\begin{aligned}\frac{\partial\log\left(b_j^{(i)}(\varphi_t)\right)}{\partial\tilde{\sigma}_{jrm}^{(i)}} = \frac{1}{b_j^{(i)}(\varphi_t)}c_{jr}^{(i)}\left|\Sigma_{jr}^{(i)}\right|^{-\rho/2}\left(\left(\frac{\varphi_{tl}-\mu_{jrl}^{(i)}}{\sigma_{jrl}^{(i)}}\right)^2 - 1\right)... \\ \exp\left(-\frac{1}{2}\sum_{l=1}^{\rho}\left(\frac{\varphi_{tl}-\mu_{jrl}^{(i)}}{\sigma_{jrl}^{(i)}}\right)^2\right)\end{aligned} \tag{A.29}$$

Finalmente,

$$\begin{aligned}\tilde{\sigma}_{jrm}^{(i)}(n+1) = \tilde{\sigma}_{jrm}^{(i)}(n) + \epsilon\gamma\ell\left(d_i\left(\boldsymbol{\varphi}_n\right)\right)\left(1-\ell\left(d_i\left(\boldsymbol{\varphi}_n\right)\right)\right) \\ \frac{1}{n_\varphi}\sum_{t=1}^{n_\varphi}\delta\left(\bar{\theta}_t - j\right)c_{jr}^{(i)}\left(b_j^{(i)}\left(\varphi_t\right)\right)^{-1}\left|\Sigma_{jr}^{(i)}\right|^{-1/2}(2\pi)^{-\rho/2} \\ \exp\left(-\frac{1}{2}\sum_{l=1}^{\rho}\left(\frac{\varphi_{tl}-\mu_{jrl}^{(i)}(n)}{\sigma_{jrl}^{(i)}(n)}\right)^2\right)\left(\left(\frac{\varphi_{tm}-\mu_{jrm}^{(i)}(n)}{\sigma_{jrm}^{(i)}(n)}\right)^2 - 1\right)\end{aligned} \tag{A.30}$$

A.3.3. Actualización de los pesos de ponderación de las componentes Gaussianas

El ajuste discriminativo del parámetro c partiendo de la definición (1.34), esta dado por:

$$\tilde{c}_{jr}^{(i)}(n+1) = \tilde{c}_{jr}^{(i)}(n) - \varepsilon \left.\frac{\partial \ell_i(\varphi_n;\lambda)}{\partial \tilde{c}_{jr}^{(i)}}\right|_{\lambda=\lambda_n} \tag{A.31}$$

Por regla de la cadena,

$$\frac{\partial \ell_i(\varphi_n;\lambda)}{\partial \tilde{c}_{jr}^{(i)}} = \frac{\partial \ell_i(d_i)}{\partial d_i}\frac{\partial d_i}{\partial \tilde{c}_{jr}^{(i)}} \tag{A.32}$$

Teniendo en cuenta el resultado mostrado en (A.10), se debe ahora desarrollar el segundo factor de la ecuación (A.32). Se Tiene,

$$\frac{\partial d_i}{\partial \tilde{c}_{jr}^{(i)}} = -\frac{\partial g_i(\varphi_n;\lambda)}{\partial \tilde{c}_{jr}^{(i)}} = -\sum_{t=1}^{n_\varphi} \frac{\partial \log\left(b_{\bar{\theta}_t}^{(i)}(\varphi_t)\right)}{\partial \tilde{c}_{jr}^{(i)}} \tag{A.33}$$

$$\frac{\partial d_i}{\partial \tilde{c}_{jr}^{(i)}} = -\sum_{t=1}^{n_\varphi} \delta\left(\bar{\theta}_t - j\right) \frac{\partial \log\left(b_j^{(i)}(\varphi_t)\right)}{\partial \tilde{c}_{jr}^{(i)}} \tag{A.34}$$

Ahora,

$$\begin{aligned}\frac{\partial \log\left(b_j^{(i)}(\varphi_t)\right)}{\partial \tilde{c}_{jr}^{(i)}} = {} & \frac{1}{b_j^{(i)}(\varphi_t)} c_{jr}^{(i)} \left|\Sigma_{jr}^{(i)}\right|^{-\rho/2} \exp\left(-\tfrac{1}{2}\sum_{l=1}^{\rho}\left(\frac{\varphi_{tl}-\mu_{jrl}^{(i)}}{\sigma_{jrl}^{(i)}}\right)^2\right)\ldots \\ & \frac{\partial}{\partial \tilde{c}_{jr}^{(i)}}\left(\frac{\exp\left(\tilde{c}_{jr}^{(i)}\right)}{\sum_{l=1}^{M}\exp\left(\tilde{c}_{jl}^{(i)}\right)}\right)\end{aligned} \tag{A.35}$$

$$\frac{\partial}{\partial \tilde{c}_{jr}^{(i)}}\left(\frac{\exp\left(\tilde{c}_{jr}^{(i)}\right)}{\sum_{l=1}^{M}\exp\left(\tilde{c}_{jl}^{(i)}\right)}\right) = c_{jr}^{(i)}\left(1-c_{jr}^{(i)}\right) \tag{A.36}$$

Finalmente,

$$\begin{aligned}\tilde{c}_{jr}^{(i)}(n+1) = {} & \tilde{c}_{jr}^{(i)}(n) + \epsilon\gamma\ell\left(d_i(\boldsymbol{\varphi}_n)\right)\left(1-\ell\left(d_i(\boldsymbol{\varphi}_n)\right)\right) \\ & \tfrac{1}{n_\varphi}\sum_{t=1}^{n_\varphi}\delta\left(\bar{\theta}_t - j\right)\left(b_j^{(i)}(\varphi_t)\right)^{-1}\left|\Sigma_{jr}^{(i)}\right|^{-1/2}(2\pi)^{-\rho/2} \\ & \exp\left(-\tfrac{1}{2}\sum_{l=1}^{\rho}\left(\frac{\varphi_{tl}-\mu_{jrl}^{(i)}(n)}{\sigma_{jrl}^{(i)}(n)}\right)^2\right) c_{jr}^{(i)}\left(1-c_{jr}^{(i)}\right)\end{aligned} \tag{A.37}$$

Apéndice B

Evaluación del rendimiento de los sistemas de detección automática de patologías voz

B.1. Introducción

[1] En esta sección se describen algunos métodos, usados en diferentes áreas de la tecnología del habla, para presentar y comparar los resultados de los experimentos de forma que se puedan elegir los mejores. Los conceptos expuestos, aunque particularizados para la detección de patologías, son aplicables directamente a cualquier tarea genérica de detección y, con algunas excepciones, a tareas más amplias de clasificación.

B.2. Obtención de los resultados de un detector automático

El objetivo último de un detector automático de patología vocal (y en general de cualquier clasificador) es proporcionar una decisión tajante acerca de si la grabación de voz que se le presenta a la entrada es patológica o no. Además es conveniente que proporcione alguna medida cuantitativa del grado de seguridad que ofrece tal decisión. Estos requisitos implican calcular la probabilidad de que la clase real sea la pronosticada a partir del fichero de voz desconocido. Cuando se conoce la distribución de probabilidad subyacente de los datos del problema, la solución óptima (la que minimiza el riesgo cometido al tomar una decisión) se obtiene mediante la teoría de decisión de Bayes. Aunque en nuestro caso desconocemos la distribución estadística exacta de las voces patológicas y normales, podemos emplear métodos no parámetricos para estimar estas distribuciones a partir de

[1] Este apéndice describe la metodología de evaluación desarrollada por el Ing. Nicolás Sáenz Lechón, como parte de su investigación para la obtención del Diploma de Estudios Avanzados del programa de doctorado "Tecnologías de la Información y las Comunicaciones", de la Universidad Politécnica de Madrid, España [66].

los patrones conocidos. Por ello, se repasan a continuación algunos conceptos básicos de la teoría de Bayes.

B.2.1. Teoría de la decisión de Bayes

Las posibles clases del problema (el estado de la naturaleza) están definidas de la siguiente forma: ω_0 indicará voz patológica y ω_1 voz normal. La probabilidad a priori de que una voz sea patológica será $P(\omega_0)$ y de que sea normal será $P(\omega_1)$. La suma de ambas probabilidades es 1. Si sólo conociéramos las probabilidades a priori, diríamos que una voz desconocida es patológica si $P(\omega_0) > P(\omega_1)$ y normal en caso contrario.
Pero se supone que de la voz desconocida tenemos alguna medida (un vector de parámetros $\mathbf{x}$) que nos ayude a mejorar la clasificación. Este vector toma distintos valores para cada voz, que se expresa en términos de probabilidad. Considérese que $\mathbf{x}$ es una variable aleatoria continua, de dimension d, cuya función densidad de probabilidad depende del tipo de voz, expresada como $p(\mathbf{x}|\omega_\mathbf{j})$. A este término se le denomina verosimilitud (*likelihood* en inglés) de la clase ω_j respecto a $\mathbf{x}$, porque, si las demás cosas permanecieran constantes, la clase ω_j que obtiene un $p(\mathbf{x}|\omega_\mathbf{j})$ mayor es mas verosímil que sea la verdadera. Tenemos por tanto una verosimilitud $p(\mathbf{x}|\omega_\mathbf{0})$ para la voz patológica y otra $p(\mathbf{x}|\omega_\mathbf{1})$ para la voz normal. Según el teorema de Bayes:

$$P\left(\omega_j|\mathbf{x}\right) = \frac{p\left(\mathbf{x}|\omega_j\right) P\left(\omega_j\right)}{p\left(\mathbf{x}\right)} \tag{B.1}$$

El término $P\left(\omega_j|\mathbf{x}\right)$, denominado probabilidad *a posteriori*, representa la probabilidad de que la voz desconocida pertenezca a la clase ω_j dada la medida $\mathbf{x}$. El termino $p(\mathbf{x})$, denominado *evidencia*, puede verse como un factor de escala para que las probabilidades a posteriori sumen 1.

$$p\left(\mathbf{x}\right) = p\left(\mathbf{x}|\omega_0\right) P\left(\omega_0\right) + p\left(\mathbf{x}|\omega_1\right) P\left(\omega_1\right) \tag{B.2}$$

A partir de la ecuación (B.1) podemos por tanto establecer la regla de decisión de Bayes, que garantiza que se minimiza la probabilidad de error. Decidiremos que una voz desconocida pertenece a la clase voz patológica si:

$$P\left(\omega_0|\mathbf{x}\right) > P\left(\omega_1|\mathbf{x}\right) \tag{B.3}$$

O lo que es igual:

$$p\left(\mathbf{x}|\omega_0\right) P\left(\omega_0\right) > p\left(\mathbf{x}|\omega_1\right) P\left(\omega_1\right) \tag{B.4}$$

Mediante la teoría bayesiana podemos además incluir factores de coste por tomar determinadas decisiones, de forma que la decisión final sea la que menor riesgo implique.
Definimos la acción α_0 a decidir que el verdadero estado de la naturaleza es voz patológica ω_0 y la acción α_1 corresponde a decidir ω_1. Sea γ_{ij} el coste o pérdida por decidir ω_i cuando en realidad es ω_j. Tenemos entonces que el riego condicional $R(\alpha_i|\mathbf{x})$ de tomar la acción i cuando se obtiene la medida bfx para cada clase es:

$$\begin{aligned} R\left(\alpha_0|\mathbf{x}\right) &= \gamma_{00} P\left(\omega_0|\mathbf{x}\right) + \gamma_{01} P\left(\omega_1|\mathbf{x}\right) \\ R\left(\alpha_1|\mathbf{x}\right) &= \gamma_{10} P\left(\omega_0|\mathbf{x}\right) + \gamma_{11} P\left(\omega_1|\mathbf{x}\right) \end{aligned} \tag{B.5}$$

La regla de decisión de Bayes o de mínimo riesgo es la que decide la clase ω_0 cuando $R(\alpha_0|\mathbf{x}) < R(\alpha_1|\mathbf{x})$ y la clase ω_1 en caso contrario. Esta regla se puede reescribir de la siguiente forma: elegir ω_0 si

$$\frac{p(\mathbf{x}|\omega_0)}{p(\mathbf{x}|\omega_1)} > \frac{\gamma_{01}-\gamma_{11}}{\gamma_{10}-\gamma_{00}}\frac{P(\omega_1)}{P(\omega_0)} \tag{B.6}$$

Esta forma de ver la regla de decision de Bayes nos permite decidir la clase ω_0 si el cociente de verosimilitudes supera un valor umbral que es independiente de la observación $\mathbf{x}$.

B.2.2. Obtención de las salidas del detector automático de patología

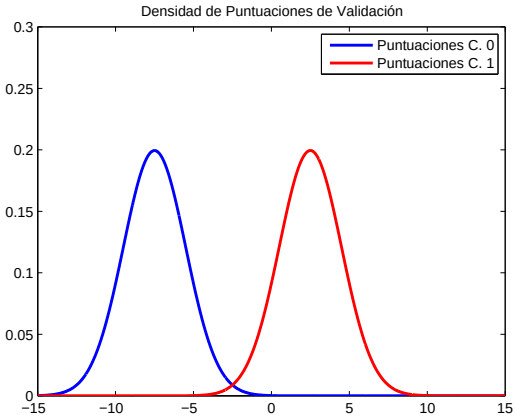
Como ya se ha dicho, la salida del detector debe ofrecer la probabilidad de que la clase pronosticada sea la correcta, dado un segmento o un fichero de voz. Gracias a la regla de decisión de Bayes, se puede calcular esa probabilidad a posteriori por medio del cociente de verosimilitudes. Lo que se pretende es estimar el cociente de verosimilitudes para cada valor posible de $\mathbf{x}$, utilizando para ello los ficheros de voz disponibles en la base de datos. A este proceso de crear un modelo se lo denomina *entrenamiento* del sistema. Dependiendo del tipo de sistema que escojamos (redes neuronales, modelos estadísticos, etc.), el método de entrenamiento variará, pero el objetivo final será el mismo. Una vez entrenado, dado un vector de entrada $\mathbf{x}$, el detector ofrecerá una medida de este cociente o *puntuación* (*score* en inglés) y se comparará con un valor umbral Λ para tomar la decisión definitiva de a qué clase pertenece el segmento o el registro de voz. Este umbral depende exclusivamente de las probabilidades a priori de cada una de las clase de voz y del coste que consideramos que conlleva tomar cada decisión.

$$\Lambda = \frac{\gamma_{01}-\gamma_{11}}{\gamma_{10}-\gamma_{00}}\frac{P(\omega_1)}{P(\omega_0)} \tag{B.7}$$

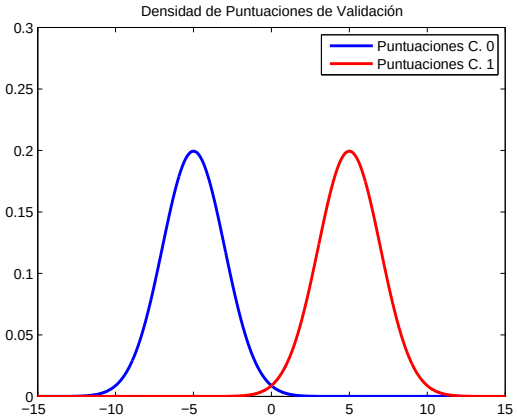
Si no se conocen las probabilidades a priori, se puede otorgar el valor arbitrario de 0.5 a cada una. Podría pensarse en utilizar la proporción de ficheros de cada clase presentes en la base de datos, pero no es válido para analizar voces desconocidas, puesto que la base de datos no refleja la realidad del problema en estudio.
En cuanto a los demás términos, es habitual considerar que tanto γ_{00} como γ_{11} no suponen coste alguno (implican decidir una clase cuando esa clase es la auténtica). Por lo que generalmente los costes que hay que tener en cuenta se limitan a los términos γ_{01} y γ_{10}. El primero representa el coste asociado a decidir que la voz es patológica cuando en realidad es normal y el segundo corresponde al coste de decidir que una voz es normal cuando en realidad es patológica. Como desde el punto de vista clínico es más importante no perderse ningún paciente enfermo que efectuar pruebas más exhaustivas a un paciente que realmente esté sano, el valor de este último término suele ser mayor que el primero. En cualquier caso, el valor del umbral de decisión debe fijarse buscando un compromiso entre ambos costes (ver apartado B.5).

Normalización de las puntuaciones

Convencionalmente en la evaluación de sistemas de reconocimiento automático, se realizan diferentes pruebas ya sea para comparar los resultados obtenidos mediante técnicas diferentes o sobre una misma técnica para evaluar su capacidad de generalización (ver sección B.4).
Si se considera un conjunto de puntuaciones provenientes de diferentes evaluaciones de la misma técnica, es decir, puntuaciones de ambas clases obtenidas para diferentes ejecuciones del algoritmo, es posible que el rango de valores cambie notablemente entre evaluaciones. Este hecho no implica que los resultados independientes de cada prueba sean muy diferentes en cuanto a tasa de reconocimiento se refiere (Ver Figura B.2.2). Sin embargo, cuando se tiene en cuanta todo el conjunto de puntuaciones de las diferentes pruebas, para proporcionar alguna medida global del desempeño del sistema, es posible que no refleje resultados coherentes con los obtenidos en las pruebas anteriores. A partir de la Figura



(a) Puntuaciones obtenidas en la primer prueba



(b) Puntuaciones obtenidas en la segunda prueba

Figura B.1: Puntuaciones obtenidas en diferentes ejecuciones de un mismo algoritmo

B.2.2 es posible observar la diferencia existente entre los umbrales óptimos de clasificación, para las dos pruebas ejemplo consideradas. Un umbral común de clasificación para

el conjunto completo de puntuaciones entregará un tasa de acierto del sistema diferente al promedio de las tasas de acierto individuales.
Una forma de corregir este problema, es normalizar las puntuaciones de cada una de las pruebas a un intervalo común [01], de tal manera que no se presenten este tipo de inconsistencias. Este tipo de normalizaciones es posible realizarlas empleando transformadas sigmoidales o logísticas [67]. La transformación consiste en obtener valores reales y pertenecientes al rango [01] mediante la aplicación de la siguiente expresión:

$$f(x) = \frac{1}{1 + e^{-(w_0 + wx)}} \in [0, 1] \tag{B.8}$$

donde,

$$w_0 = \frac{\mu_1^2 - \mu_0^2}{2\sigma^2} \qquad w = \frac{\mu_0 - \mu_1}{\sigma^2} \tag{B.9}$$

donde μ_0 y μ_1 son las medias de las puntuaciones para la clase 0 y para la clase 1 respectivamente. Esta transformación requiere asumir que los valores de las puntuaciones siguen una distribución gaussiana, con varianza común σ^2 [60]. Debido a que en la práctica la dispersion no tiene porqué ser igual, se puede usar el valor de $\sigma = 0{,}5\,(\sigma_0 + \sigma_1)$, donde σ_0 y σ_1 son las estimaciones de las correspondientes desviaciones típicas individuales.

B.3. Presentación de los resultados

Para mostrar los resultados del detector de patología vocal (y en general, de cualquier sistema de clasificación de patrones) se utiliza la llamada *matriz de contingencia o de confusión*, que recoge el número de aciertos y fallos del sistema para cada una de sus posibles salidas (las clases en que se divide el problema).
El aspecto genérico de una matriz de confusión con dos clases se muestra en la figura B.2. Las casillas con fondo blanco deben rellenarse con el número de patrones de cada clase que el detector utilizado ha clasificado del conjunto de datos. Estos valores pueden darse en valor absoluto o porcentual.
De acuerdo con esta matriz y tomando como referencia una de las clases (normalmente la que interesa detectar; en este caso cogemos la clase 0), se definen los siguientes términos:

- *Detección correcta o aceptación verdadera* (TP, *true positive*): el número (o porcentaje) de patrones de clase 0 que el clasificador asigna correctamente como pertenecientes a la clase 0.

- *Falso rechazo* (FN, *false negative*): el número (o porcentaje) de patrones de clase 0 que el clasificador asigna incorrectamente como pertenecientes a la clase 1.

- *Falsa aceptación* (FP, *false positive*):el número (o porcentaje) de patrones de clase 1 que el clasificador asigna incorrectamente como pertenecientes a la clase 0.

- *Rechazo correcto o verdadero* (TN, *true negative*):el número (o porcentaje) de patrones de clase 1 que el clasificador asigna correctamente como pertenecientes a la clase 1.

		Clase real	
		Clase 0	Clase 1
Clase estimada por el clasificador	Clase 0	TP	FP
	Clase 1	FN	TN

Figura B.2: Aspecto general de una matriz de confusión o contingencia con dos clases.

Nótese que cuando los valores se representan en porcentaje, $TP+FN = 100$ y $FP+TN = 100$. A partir de esos valores se calcula:

- *Tasa de acierto o eficiencia* (CCR, *Correct Classification Rate*): es la proporción de patrones correctamente clasificados por la red.

$$CRR = \frac{TP + TN}{TP + FN + FP + TN} \tag{B.10}$$

- *Tasa de error* (ER, *Error Rate*): es el complementario a la tasa de acierto, es decir, la proporción de patrones mal clasificados.

$$ER = 1 - CCR = \frac{FN + FP}{TP + FN + FP + TN} \tag{B.11}$$

En el caso ideal, la tasa de acierto debe ser del 100 % y la tasa de error del 0 %. Aunque ambas medidas pueden usarse indistintamente, en tareas donde la tasa de aciertos es muy alta se utiliza a menudo la tasa de error como indicador único. Si el número de patrones de las clases 0 y 1 no es el mismo, las tasas de acierto y de error no reflejan realmente el funcionamiento del sistema. Si por ejemplo tuviéramos 90 patrones de clase 0 y 10 patrones de clase 1, un detector que siempre diera como salida "clase 0" tendría un 90 % de acierto total, pero sin embargo fallaría todos los patrones de clase 1. Para corregir estas medidas, se emplean estos otros parámetros:

- *Sensibilidad* (S) da una indicación de la capacidad del sistema para detectar los patrones de la clase de referencia. Cuando los valores se representan en porcentaje, la sensibilidad coincide con TP.

$$S = \frac{TP}{TP + FN} \tag{B.12}$$

- *Especificidad* (E) da una idea de la capacidad del sistema para rechazar los patrones que no pertenecen a la clase de referencia. Cuando los valores se representan en porcentaje, la especificidad coincide con TN.

$$E = \frac{TN}{TN + FP} \tag{B.13}$$

En el caso ideal, S y E deben ser 1 (o el 100 % si se miden en porcentaje).

B.3.1. Medida de la tasa de acierto total en base a fichero y a segmentos

Si el sistema de detección de patología se basa en parámetros acústicos medidos a partir de todo el registro de la voz (habitualmente llamados parámetros "a largo plazo"), la tasa de acierto del sistema tal como se ha definido anteriormente representa el número de ficheros de voz correctamente clasificados. Sin embargo, cuando la parametrización se basa en fragmentos o segmentos temporales de la señal acústica (se habla entonces de parámetros "a corto plazo", típicamente calculados en ventanas del orden de decenas de milisegundos), la tasa de acierto recoge el número de tales segmentos correctamente clasificados. Esta medida no tiene por qué coincidir con el número de ficheros acertados realmente.

Para obtener una medida de la tasa de acierto de fichero en este último caso, se pueden utilizar diferentes estrategias. La más sencilla consiste en pronosticar la clase a la que pertenece el fichero basándose en el número de segmentos de ese fichero que pertenecen a cada clase. Se establece un umbral por el que si un porcentaje determinado de los segmentos que pertenecen al fichero han sido asignados a una clase el fichero también se considera que pertenece a esa clase. Este umbral varía típicamente entre el 51 % (si la mitad más uno de los segmentos corresponden a una clase, el fichero se postula de esa misma clase y por tanto el acierto o fallo del sistema se calcula de acuerdo a esa suposición) y valores más altos, que exigen un mayor consenso basado en los segmentos para considerar un fichero de una clase determinada.

Otra posibilidad para asignar un fichero a una clase es calculando una puntuación única para cada fichero a partir de las puntuaciones individuales de todos los vectores que lo componen [68]. Para ello se calcula la verosimilitud de cada una de las clases ω_j dado el fichero $\mathbf{X}$, compuesto por los vectores $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T$ (Ec.) A partir del cociente de estas verosimilitudes, se obtienen las puntuaciones y se establece un nuevo valor de umbral que otorgue la decisión final.

$$p\left(\mathbf{X}|\omega_j\right) = \sqrt[T]{\prod_{i=1}^{T} p\left(\mathbf{x}_i|\omega_j\right)} \tag{B.14}$$

Cuando la función de verosimilitud está dada en logaritmo, la ecuación (B.14), se puede expresar como [25]

$$\log p\left(\mathbf{X}|\omega_j\right) = \frac{1}{T}\sum_{i=1}^{T} \log p\left(\mathbf{x}_i|\omega_j\right) \tag{B.15}$$

B.4. Estimación de la capacidad de generalización del modelo

Una vez que se ha entrenado el sistema de detección de patología vocal, se puede emplear para predecir el tipo de voz de una grabación desconocida. Un punto fundamental en este momento es determinar qué grado de confianza merecen las decisiones del detector, ahora que tiene que trabajar con voces que no han sido evaluadas anteriormente por un especialista médico. El hecho de obtener una tasa de acierto determinada para un conjunto de

N patrones conocidos no garantiza que con otro conjunto diferente los resultados vayan a ser los mismos. Si la prueba se repitiera por ejemplo con 20 conjuntos de datos distintos, se obtendrían otras tantas tasas de acierto diferentes, aunque fuera de esperar que se pareciesen bastante entre sí.

El cálculo exacto del grado de error cometido por el detector es imposible, pero se puede obtener una estimación del error de clasificación del modelo (o lo que es igual, de la capacidad de generalización que posee) a partir de los datos utilizados para el aprendizaje supervisado. A este paso se lo conoce en reconocimiento de patrones como *validación o estimación de la generalización* del modelo.

Además de obtener un valor estimado de la tasa de acierto, conviene añadir además un rango de valores alrededor del cual se puede encontrar el valor real, para una probabilidad dada. Al rango de valores se le llama *intervalo de confianza* de la medida (Figura B.3). A la probabilidad con la que la tasa real se encuentra dentro del intervalo se le denomina nivel de confianza, que se expresa habitualmente mediante el parámetro α. Para expresarlo en porcentaje hay que hacer $100(1-\alpha)$. Valores típicos de α son 0,05 (95 %) o 0,01 (99 %). Es de destacar que cuanto mayor sea el nivel de confianza requerido, mayor se hará el intervalo de confianza. Dado un valor de tasa de acierto CCR del 90 % y un intervalo de

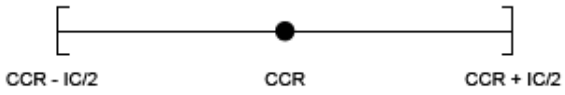


Figura B.3: Intervalo de confianza de una medida.

confianza para esa medida de ± 1 %, con un nivel de confianza α del 0,05, la interpretación de ese valor es: "Se Tiene una confianza del 95 % en que el valor real de la tasa de acierto del modelo está dentro del intervalo $89-91$ %".

Otros conceptos que conviene tener en cuenta a la hora de obtener una medida de la precisión del modelo son el *sesgo* (bias) y la *varianza*. El sesgo de un estimador es la diferencia entre su valor verdadero (generalmente desconocido) y su valor esperado. Cuanto menor sea el sesgo, mayor precisión tendrá la estimación en promedio. La varianza está relacionada con cuánto cambia el valor de la estimación cuando cambiamos el conjunto de datos usado para obtenerla. El error de clasificación de un modelo es función de ambos estadísticos y hay un compromiso entre ellos. Si el método de entrenamiento del modelo usado puede adaptarse fácilmente a los datos de entrenamiento (por ejemplo porque tenga muchos parámetros libres), el sesgo tenderá a ser menor, a costa de una mayor varianza. En clasificación de patrones, la varianza es más importante que el sesgo: en la práctica, si se mantiene la varianza baja no hay que preocuparse demasiado por el sesgo de la medida [3]. Y en general, cuantos más datos de entrenamiento se tengan, menor será la varianza.

A continuación se presentan distintas formas posibles de realizar la validación del modelo y, en los casos en que es posible, obtener los intervalos de confianza.

B.4.1. Validación basada en estadísticos de la muestra (del conjunto de datos)

La teoría estadística proporciona varios estadísticos simples para el error de generalización en modelos lineales bajo ciertas condiciones de la muestra. Estos estadísticos también se pueden emplear como estimaciones groseras del error de generalización en modelos no lineales cuando se tiene un conjunto "grande" de entrenamiento. La adaptación de los estadísticos para la no linealidad requiere mayores cálculos y no siempre es posible hacerlo en todas las técnicas de clasificación.
La ecuación (B.16) recoge un estadístico utilizado en tecnología del habla [69]. Si se realiza la prueba del modelo entrenado con un conjunto de N patrones y se obtiene una tasa de acierto p, se puede obtener el intervalo de confianza como:

$$IC = p \pm z_{\alpha/2}\sqrt{\frac{p\,(1-p)}{N}} \tag{B.16}$$

El valor de z se obtiene a partir de la función de distribución normal estándar, en función del nivel de confianza α requerido. Para un valor de α de 0,05 (95 % de confianza), z vale 1,96.
Como se puede observar, la anchura del intervalo depende del nivel de confianza deseado (si se requiere mayor confianza, el intervalo se hará mayor) y del número de patrones utilizados para realizar la prueba. Cuando las medidas se obtienen a corto plazo, el factor N podría referirse tanto al número total de patrones del conjunto de prueba como al número de ficheros empleados para generar dichos patrones. Es evidente que cuanto mayor sea N, menor será el intervalo para un nivel de confianza dado. El problema aquí es que hay que dar por supuesta la independencia estadística entre los patrones pertenecientes a un mismo fichero. Si se considera N como el número de ficheros de voz realmente disponibles (y por tanto p también debe ser medida en base a los ficheros), el problema es que se necesita entonces una cantidad elevada de estos.
Respecto a esto, conviene tener en cuenta la "regla de los 30" de Doddington [70], que dice: "Para tener una confianza del 90 % que la verdadera tasa de error está dentro del ±30 % de la tasa de error observada, debe haber al menos 30 errores". Supóngase que se persigue, por ejemplo, una tasa de falsa aceptación menor del 5 % y un falso rechazo menor del 10 %. 30 errores al 5 % de falsa aceptación suponen un total de 600 grabaciones de voces normales. De igual forma, 30 errores el 10 % de falso rechazo suponen 300 grabaciones de voces patológicas. La moraleja más extendida que se adopta en tecnología del habla, referente a estas cuestiones, consiste en dar por supuesta la independencia de los patrones y no tomar demasiado seriamente las afirmaciones sobre la significación estadística de los resultados.

B.4.2. Validación por resustitución

En este caso, el modelo se entrena con todos los patrones disponibles en la base de datos. Posteriormente se clasifican los patrones con el modelo ya entrenado y se obtiene la proporción de patrones correcta e incorrectamente clasificados. El problema de este estimador es que se calcula empleando el mismo conjunto de datos usado para construir el modelo,

por lo que proporciona un estimador de la bondad del modelo sesgado y optimista (la estimación es mejor que la realidad). La estimación de la generalización del modelo es el porcentaje de ficheros correctamente clasificados. Este método no permite obtener un intervalo de confianza para la tasa de acierto estimada.

B.4.3. Validación por partición de la muestra (split-sample o holdout)

El método más usado para estimar el error de generalización en reconocimiento de patrones es reservar parte de los datos como un conjunto de prueba, que no puede ser usado de ningún modo durante el entrenamiento. El conjunto de prueba debe ser una muestra representativa de los casos sobre los que se quiere generalizar. Después del entrenamiento, se evalúa el modelo con los datos de prueba. La estimación de la generalización del modelo es el porcentaje de ficheros del conjunto de prueba correctamente clasificados.
La principal desventaja de la validación por partición de la muestra es que reduce mucho la cantidad de datos de entrenamiento y puede no ser factible si se dispone de pocos datos. La estimación del rendimiento obtenida tiende a ser pesimista (a menos que el número de datos N sea grande) y también poco fiable, porque su valor depende de la partición elegida [71]. Cuantos más patrones se reservan para el conjunto de prueba, mayor será el sesgo de la estimación; a cambio, cuanto menor sea el conjunto de prueba, mayor será su intervalo de confianza [72].
Se puede aumentar la fiabilidad promediando sobre todas las posibles particiones de la muestra de tamaño fijo. Se divide la muestra en dos conjuntos con patrones elegidos al azar y mutuamente excluyentes y se obtiene una estimación del rendimiento. Se repite k veces el mismo proceso y la estimación final será el promedio de las estimaciones parciales. La desviación típica puede obtenerse como la desviación típica de las estimaciones parciales. Este método se conoce como "data shuffling" [73] o "submuestreo aleatorio" [72]. La utilización de los datos sigue siendo poco eficiente, puesto que sólo se usa una parte para entrenar en cada iteración. El resultado sigue siendo excesivamente pesimista.
Valores típicos del tamaño del conjunto de prueba están entre $^1/_2$ y $^1/_3$ del total de datos disponibles.

B.4.4. Estimación por validación cruzada

Para remediar los inconvenientes anteriores hay diversas técnicas, denominadas de "validación cruzada". La validación cruzada es una mejora del método de validación anterior, que permite usar todos los datos disponibles para el entrenamiento y aún así obtener un estimador del error de generalización menos sesgado. Su desventaja es que estos métodos exigen entrenar el modelo varias veces, con el coste computacional que ello conlleva. Hay varias modalidades de validación cruzada.

K-fold: El conjunto de datos se divide de forma aleatoria en k subconjuntos independientes de aproximadamente igual tamaño. Se efectúa el entrenamiento y prueba del modelo k veces, dejando fuera del entrenamiento un subconjunto diferente cada vez. Con este subconjunto se valida el funcionamiento del modelo entrenado con los $k-1$ subconjuntos

restantes. La estimación de la generalización del modelo es el promedio de las tasas de clasificación obtenidas con cada uno de los subconjuntos de prueba. Este método tiene menor sesgo que el de partición de la muestra anterior (aunque depende del valor de k, del número de datos disponibles y de la dimensión de los mismos) La estimación es pesimista, aunque mejora según se eligen más subconjuntos [72]. Valores típicos de k suelen ser del orden de 5 a 20. Una variante de este método es la validación cruzada *estratificada*, donde los subconjuntos contienen aproximadamente la misma proporción de patrones de cada clase que el conjunto de datos original.

Leave-one-out: Es un *k-fold* extremo donde k es igual al número total de datos disponibles. Se entrena el modelo dejando fuera un solo fichero, que se usa para validar el resultado. El proceso se repite hasta utilizar todos y cada uno de los ficheros para validación. El promedio de todas las validaciones es la estimación final del rendimiento. Este método también es conocido en estadística como *jackknife* [3] o herramental [74].
La validación cruzada es claramente superior para conjuntos de datos pequeños a la validación por partición del conjunto de datos. Al término del proceso, se han aprovechado todos los datos disponibles para entrenar y validar el modelo. El estimador obtenido no está sesgado puesto que en cada resultado parcial no se usan los mismos datos para entrenar que para probar. Sin embargo puede tener más varianza que otros métodos, lo que a veces es peor.

B.4.5. Bootstrapping

Es una mejora de la validación cruzada que a menudo proporciona mejores estimaciones del error de generalización, al coste de un mayor tiempo de cálculo todavía. El término "bootstrap" (literalmente, correa de bota) proviene de los relatos del escritor alemán R.E. Raspe "Las aventuras del Barón de Munchausen", en las que el héroe era capaz de subirse a su caballo tirando de las correas de sus botas de montar [73]. Peña [74] lo traduce por "método autosuficiente".
El método consiste en calcular la varianza de la estimación considerando el conjunto de datos disponible como si fuera la población total del problema y obtener muestras aleatorias a partir de ella según el método de Montecarlo [74]. Dado un conjunto de datos de entrenamiento de tamaño n, el proceso para obtener la estimación autosuficiente consiste en crear un número B de subconjuntos de entrenamiento a partir del original, con igual tamaño n, generados extrayendo elementos de forma aleatoria y reintroduciéndolos antes de cada extracción. Es decir, que en cada conjunto puede haber patrones repetidos. Con cada uno de estos B conjuntos de tamaño n se entrena un modelo y se prueba con el conjunto de prueba, formado por los patrones que no han entrado en el conjunto de entrenamiento. El estimador final será el promedio de los B estimadores obtenidos.
Ha sido aplicado a diferentes métodos de clasificación, como árboles de decisión [72] o redes neuronales [75]. El principal inconveniente de este método es que es computacionalmente intensivo, requiriendo una gran cantidad de repeticiones. Valores típicos del número de repeticiones, desde 200 en adelante. A cambio, una ventaja respecto al *leave-one-out* es que cuantas más repeticiones se hagan, mayor precisión tendrá la estimación, mientras que con este segundo método, una vez repetido tantas veces como datos, ya no aumenta

la precisión.

B.4.6. Precaución sobre los métodos de validación

Si se emplea cualquiera de los métodos anteriores (validación por partición de la muestra, validación cruzada, *bootstrapping*), tal como han sido descritos, para seleccionar el mejor modelo posible de entre varios, la estimación del error de generalización de ese modelo será optimista. Si se entrenan varios modelos usando un *conjunto de entrenamiento* y se usa un segundo conjunto de datos (*conjunto de validación*) para decidir qué modelo es el mejor, se debe usar un tercer conjunto (*conjunto de prueba*) para obtener una estimación no sesgada del error de generalización del modelo elegido. También puede, una vez elegido el modelo con los valores de los parámetros que funcionan mejor, entrenar de nuevo ese modelo con los conjuntos de entrenamiento y de validación juntos, y medir su capacidad de generalización con el conjunto de prueba [76].
Una última consideración al estimar el error de generalización con un conjunto de datos dado, utilizando cualquiera de los métodos descritos. Para poder dar una estimación fiable, hay que preguntarse si los datos disponibles representan adecuadamente la variabilidad de los datos para la tarea en consideración [71]. Por ejemplo, en reconocimiento de voz se necesitan miles de patrones para recoger la verdadera variabilidad de los datos de la población general. Ese es el precio a pagar por no conocer las distribuciones de probabilidad condicional subyacentes de cada clase.

B.5. Curvas de rendimiento de un detector

Las tareas de detección pueden verse como un compromiso entre dos tipos de error: detecciones fallidas (falso negativo o falso rechazo) y falsas alarmas (falso positivo o falsa aceptación). Por ejemplo, un sistema de reconocimiento de patología vocal puede fallar no detectando una enfermedad conocida o puede declarar que la ha detectado cuando no está presente en realidad. Este compromiso se refleja en la ecuación (B.7), en la que si eliminamos los términos de las probabilidades a priori y los costes asociados a elegir la opción correcta, queda:

$$\Lambda = \frac{\gamma_{01}}{\gamma_{10}} \tag{B.17}$$

En esta ecuación, γ_{01} y γ_{10} representan el coste asociado a un falsa aceptación (decidir clase 0 cuando es clase 1) y a un falso rechazo (decidir clase 1 cuando es clase 0) respectivamente. En estos casos, no es adecuado representar la capacidad del sistema mediante un único indicador numérico del rendimiento, puesto que hay varios puntos de operación posibles en función del umbral Λ elegido. El funcionamiento del sistema queda mejor representado mediante una *curva de rendimiento*. Aunque en la literatura hay varias curvas diferentes, su cálculo se basa en la representación gráfica de los valores de falsa aceptación y falso rechazo que se obtienen al variar el umbral Λ.
Para calcular la curva de falso rechazo se utilizan los cocientes de verosimilitud obtenidos con los patrones de la clase 0, con el conjunto de datos que se esté utilizando. Con estas puntuaciones se crea un histograma, que debería de estar situado en su mayor parte a la

derecha del umbral $\Lambda = 1$ (o $\Lambda = 0$ si se usan logaritmos). Este histograma (Figura B.4, color azul) se normaliza, dividiendo los valores del eje de ordenadas entre el número total de patrones. El histograma normalizado se puede interpretar como una versión discreta de la función densidad de probabilidad de la clase 0. A partir de esta función se calcula la función de distribución correspondiente, acumulando los valores desde la izquierda hacia la derecha. Para cada valor de umbral en el eje de abscisas, el eje de ordenadas ofrece la tasa de patrones de clase 0 que han sido clasificados como de clase 1 (Figura B.5, color azul). Para calcular la curva de falsa aceptación se procede de manera similar. Con las

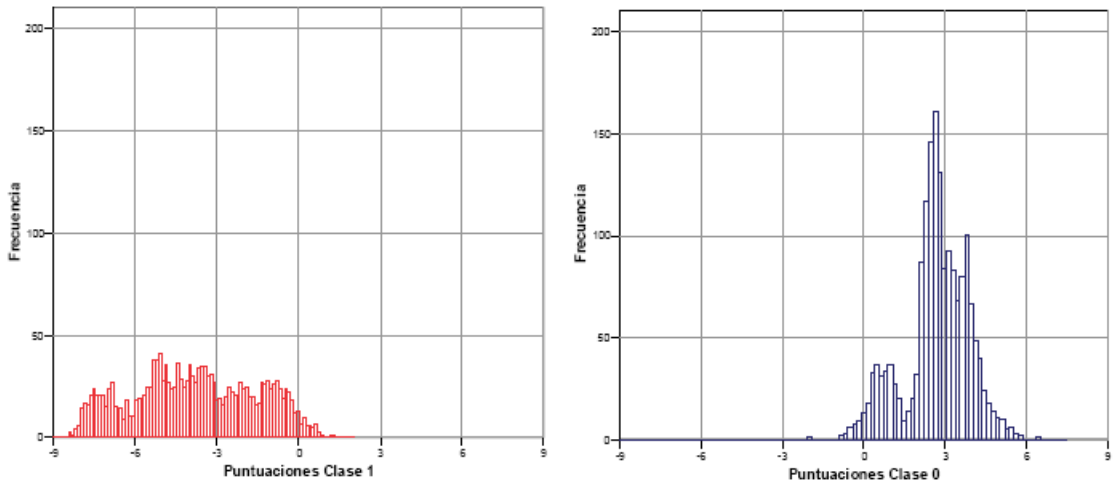


Figura B.4: Histogramas del logaritmo de las puntuaciones de clase 1 (que en este ejemplo corresponde a voz patológica) y clase 0 (voz normal), con un conjunto de datos de 1755 y 1754 patrones respectivamente. Nótese que los histogramas no están normalizados.

puntuaciones de los patrones de clase 1 se forma un histograma normalizado, que debería de estar situado a la izquierda del valor $\Lambda = 1$ (Figura B.4, color rojo). El histograma se acumula desde la derecha hacia la izquierda para calcular la función de distribución de las voces de clase 1. Para cada valor de umbral, el eje de ordenadas indica la proporción de patrones de clase 0 clasificados como clase 1 (Figura B.5, color rojo). En estas curvas se pueden señalar varios puntos de interés. El punto donde se igualan las tasas de falso acierto y falso rechazo se denomina EER (punto de *Equal Error Rate* o *Tasa de Equi-Error*). Para calcularlo, se restan las curvas, se halla su valor absoluto y el mínimo indica el EER. El punto de operación del sistema es el que corresponde al umbral elegido. Habitualmente el umbral se elige con el conjunto de datos con el que se ha entrenado el modelo y se utiliza posteriormente para la validación del sistema y la obtención de su punto de operación real.

B.5.1. Curvas DET

Las curva DET (*Detection Error Trade-off*) fue desarrollada en el Instituto Nacional de Estándares y Tecnología (NIST) de los EEUU para la evaluación de sistemas de reconocimiento de locutor [64]. Representa la tasa de falso rechazo (denominada probabilidad de fallo o *miss* en la terminología del NIST) en función de la tasa de falsa aceptación del sistema (Figura B.6). En la curva DET se presupone que la distribución de las puntua-

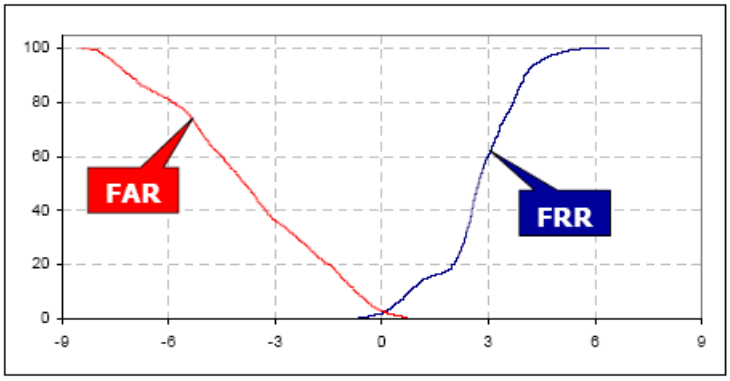


Figura B.5: Curvas de Falso Rechazo (azul, a la derecha) y Falsa Aceptación (roja, a la izquierda) correspondientes a los histogramas de la Figura B.4. En el eje de abscisas se representan los posibles umbrales de decisión con los que puede operar el sistema y en el eje de ordenadas se obtienen los valores de FR y FA correspondientes.

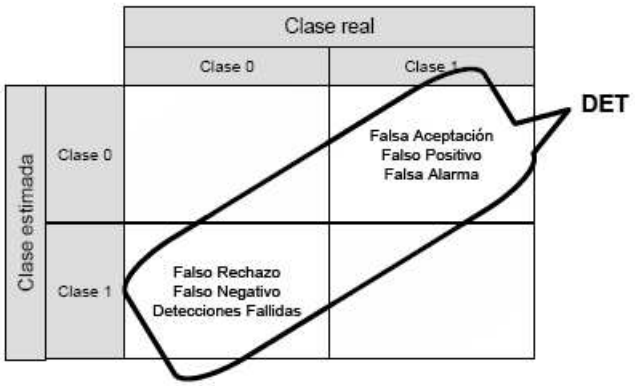


Figura B.6: Medidas representadas en una curva de tipo DET y distintas denominaciones que reciben en la literatura.

ciones de ambas clase es próxima a la normal, por lo que la escala de ambos ejes sigue esa distribución (Figura B.7). A consecuencia de ello, cuando las distribuciones reales se acercan a la normalidad, las curvas tienden a ser líneas rectas. Además, la inclinación de la curva recta está en relación con la desviación típica de las distribuciones. Si ambas desviaciones típicas son iguales, la pendiente es unidad. En la Figura B.7, el punto medio de los ejes de abscisas y ordenadas, con un valor del 50 %, se corresponde con la media de una curva normal. El valor de cada punto de un eje se corresponde con el área comprendida bajo la curva normal entre menos infinito y el punto elegido. En la parte superior y en el margen derecho se muestran los mismos valores, medidos en desviaciones típicas desde la media. La parte de la gráfica situada por encima de la diagonal $x = -y$ carece de uso, puesto que corresponde a puntos de funcionamiento de un sistema en los que la suma de las falsas aceptaciones y falsos rechazos suman más del 100 %, lo que es imposible.
En la curva DET, la diagonal $x = -y$ representa el funcionamiento de un sistema de decisión aleatorio. Esto se ilustra en la Figura B.8, donde se presentan dos supuestas distribuciones de puntuaciones pertenecientes a las clases 0 y 1. Por simplicidad se han

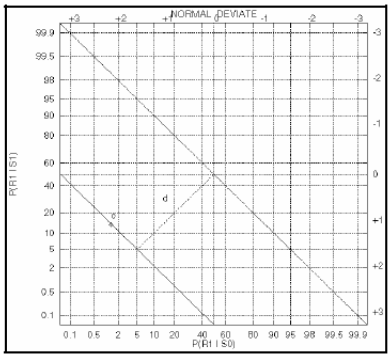


Figura B.7: Escala de la distribución normal en que se representan las curvas de tipo DET. Figura tomada de [64].

supuesto ambas distribuciones normales y con igual varianza. Se coja el umbral de decisión que se coja, la tasa de acierto global del sistema será siempre del 50 %. Evidentemente, este el peor de los casos posibles en cualquier sistema de detección automática. Cuando

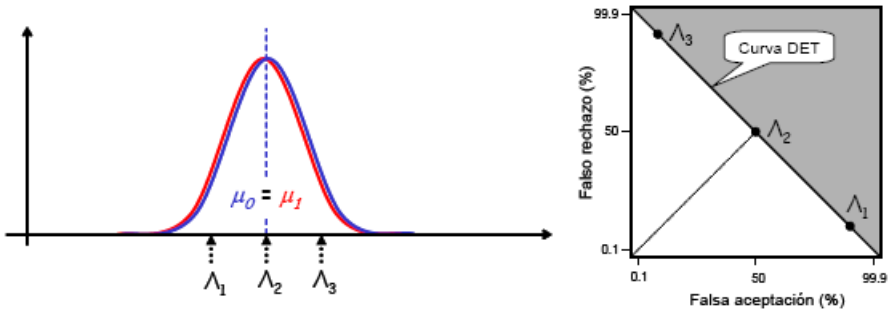


Figura B.8: Curva DET cuando las distribuciones de ambas clases están completamente solapadas (sistema aleatorio).

las distribuciones de los cocientes de verosimilitud de ambas clases no están completamente solapadas, el rendimiento del sistema mejora y la curva DET evoluciona hacia la esquina inferior izquierda de la gráfica Figura B.9. Cuanto más separadas estén ambas distribuciones, mejor será el rendimiento del detector. El punto exacto de trabajo del sistema será función del umbral Λ elegido, que a su vez es función de los costes asignados a las falsas aceptaciones y falsos rechazos. Cuando la tasa de acierto de un detector es suficientemente alta, la curva DET se acerca a la esquina inferior izquierda de la gráfica, por lo que a menudo sólo se muestra esa parte por comodidad. En la curva DET no se muestran referencias explícitas al valor de umbral elegido, puesto que su valor no tiene un significado fuera del sistema particular. De las Figuras B.8 y B.9 se puede intuir que el valor del umbral se refleja en el punto de trabajo reflejado sobre la curva (que en los ejemplos es una línea recta por ser ambas distribuciones normales). Cuanto menor es el umbral de decisión, más abajo y a la derecha estará el punto de trabajo, o lo que es lo mismo, el sistema permitirá más falsas aceptaciones a costa de restringir los falsos rechazos. Desde el punto de vista de un sistema de decisión médica, puede ser conveniente favorecer las falsas aceptaciones sobre el falso rechazo (es decir, no dar por sano a ningún

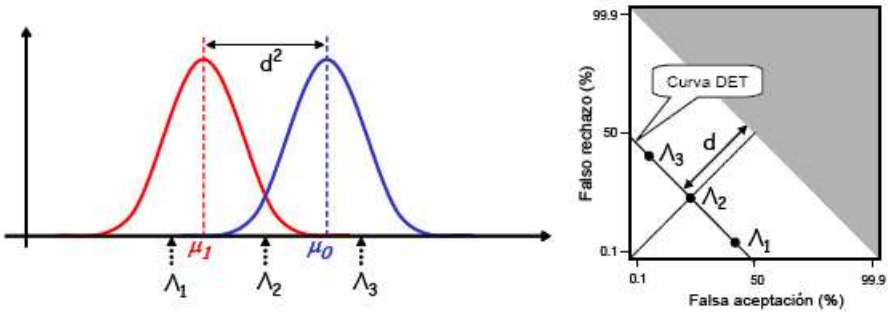


Figura B.9: Curva DET cuando las distribuciones de ambas clases están parcialmente solapadas.

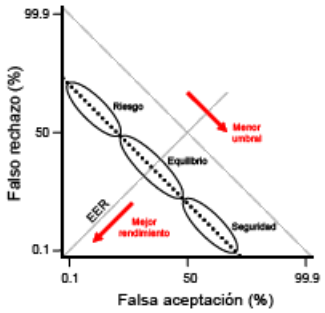


Figura B.10: Resumen del funcionamiento de la curva DET.

enfermo, a costa incluso de considerar enferma a alguna persona saludable). En la Figura B.10 el rango marcado como "seguridad" refleja esa posible zona de trabajo en la curva DET. Esto equivale a que el coste asociado a un falsa aceptación (γ_{01}) sea menor que el coste asociado a un falso rechazo (γ_{10}) en la ecuación (B.17) a la hora de fijar el umbral de decisión.

B.5.2. Curvas ROC

La curva más utilizada en la literatura médica para la toma de decisiones es la curva ROC (Característica de Operación del Receptor). Su origen se remonta a la década de 1950, en la detección de señales de radio contaminadas por ruido. En la ROC se representa la tasa de falso acierto (FA) en función de la tasa de acierto (1-FR) para diferentes valores del umbral de decisión (Figura B.11). Al igual que en la DET, la forma y posición de la ROC depende de la forma y del solapamiento de las distribuciones subyacentes de las voces patológicas y normales. Esto se observa en la Figura B.12, donde por simplicidad se han supuesto ambas distribuciones normales y de igual varianza. Una vez más, el punto de trabajo del sistema vendrá determinado por el valor de umbral Λ escogido. Se han propuesto varias medidas teóricas para reducir la curva ROC a un único indicador de la precisión del diagnóstico [63]. La más utilizada es el área bajo la curva (AUC). El área bajo la ROC puede usarse como una estimación de la probabilidad de que la anormalidad detectada permita una identificación correcta. Este índice varía entre 0,5 (no hay precisión aparente) y 1,0 (precisión perfecta) a medida que la curva ROC se mueve hacia

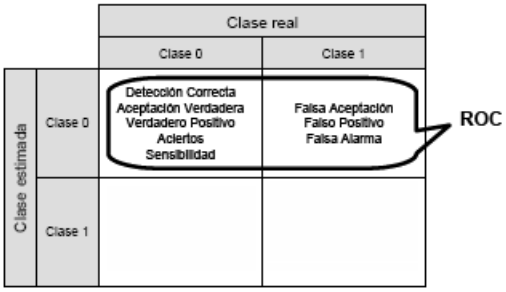


Figura B.11: Medidas representadas en una curva de tipo ROC y distintas denominaciones que reciben en la literatura.

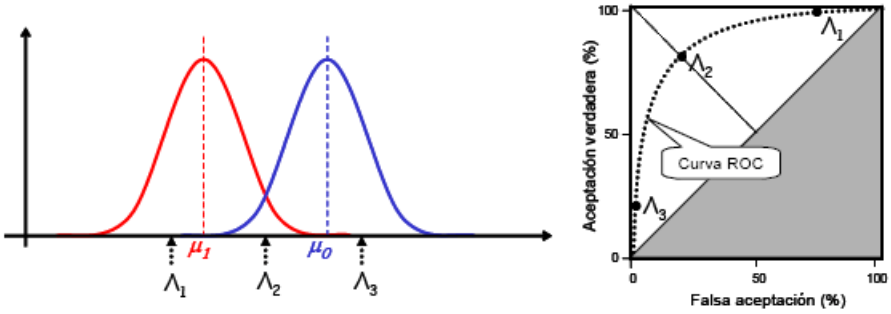


Figura B.12: Curva ROC cuando las distribuciones de ambas clases están parcialmente solapadas.

los márgenes izquierdo y superior. Cuanto mayor sea el área bajo la curva, mejor será el rendimiento del sistema.
Una cuestión importante en las curvas ROC es que ofrecen la posibilidad de comparar dos o más pruebas obtenidas con los mismos datos de forma cuantitativa [77]. A partir del área A_i bajo cada una de las curvas, se calcula el siguiente estadístico z:

$$z = \frac{A_1 - A_2}{\sigma\left(A_1 - A_2\right)} \tag{B.18}$$

En la ecuación B.18, $\sigma\left(A_1 - A_2\right)$ representa la desviación típica de la diferencia de areas, calculada mediante:

$$\sigma\left(A_1 - A_2\right) = \sqrt{\sigma^2\left(A_1\right) + \sigma^2\left(A_2\right) - 2r\sigma\left(A_1\right)\sigma\left(A_2\right)} \tag{B.19}$$

El valor de r representa la correlación inducida entre las dos áreas al realizar el estudio a partir de los mismos datos. La desviación típica de cada área se obtiene a partir de la siguiente expresión, donde n_n y n_p son el número de casos normales y patológicos respectivamente.

$$\sigma\left(A\right) = \sqrt{\frac{A\left(1-A\right) + \left(n_n - 1\right)\left(\frac{A}{2-A} - A^2\right) + \left(n_p - 1\right)\left(\frac{2A}{1+A} - A^2\right)}{n_n n_p}} \tag{B.20}$$

Si el valor de z está por encima de un valor umbral dado (típicamente 1,96), se considera que existen diferencias significativas entre ambas curvas (son diferentes con un grado de

certeza del 95 %).

En la Figura B.13 se muestra un esquema del funcionamiento de la curva ROC y sus posibles puntos de operación.

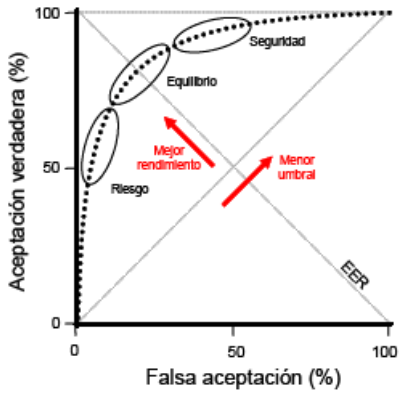


Figura B.13: Resumen del comportamiento de la curva ROC.

Apéndice C

Análisis de variables dinámicas empleando PCA

[2] Es usual que en las tareas de reconocimiento de patrones, la representación de observaciones se realice por medio de la generación de *características de tipo estático*, es decir, valores numéricos que representan algún atributo de la señal u observación y que además se asumen constantes a través de dimensiones asociadas (por ejemplo: el tiempo, el espacio, entre otras) a dicha observación. Sin embargo, existe otro tipo de características que se conocen como *de tipo dinámico*, y son valores numéricos que representan algún atributo de la señal u observación, que cambian con relación a alguna dimensión asociada; una característica dinámica se puede representar por un vector de datos para una única observación. La ventaja de las características dinámicas es que permiten representar de mejor forma el comportamiento, o dinámica de cambio propia de las señales u observaciones.
El objetivo es extender el análisis tradicionalmente estático de la técnica PCA, a un análisis de tipo dinámico, es decir, realizar el análisis sobre características dinámicas.
En este sentido, para el reconocimiento de patrones, y desde el punto de vista de teoría de la información, se desea extraer la información relevante de las variables dinámicas que representan la observación, codificarla tan eficientemente como sea posible y comparar la representación codificada de la observación contra una base de datos de patrones o modelos codificados similarmente. Una aproximación para extraer la información contenida en las variables dinámicas es, de alguna forma, capturar las variaciones presentes en un conjunto de observaciones, y utilizar esta información para codificar y comparar las observaciones. Lo anterior puede entenderse como, obtener los componentes principales de la distribución de las observaciones, o los vectores propios de la matriz de covarianza del conjunto de observaciones [78]. La idea de emplear esta forma de representación, se debe a que es natural pensar que el procedimiento desarrollado en [79] conocido como *Eigenfaces*, puede extenderse a otros tipos de objetos, por ejemplo, observaciones representadas por características de tipo dinámico en el tiempo.
El proceso enfocado hacia la reducción de características dinámicas y posterior clasificación, consiste en:

[2] Este apéndice hace parte del trabajo desarrollado por el Ing. Genaro Daza Santacoloma, en su tesis de Maestría, titulada: "Metodología de reducción de dimensión para sistemas de reconocimiento automático de patrones sobre bioseñales" [58]

1. Ordenar las características dinámicas, de tal forma que las varianzas y covarianzas se puedan estimar entre todos los puntos de representación de las variables.

2. Calcular las componentes principales del arreglo de elementos ordenados, representando las observaciones. Los vectores propios obtenidos generan la base de un subespacio que abarca la mayoría de la información dada por un conjunto de observaciones de entrenamiento.

3. Como estos vectores propios conforman una base ortonormal, pueden ser usados para proyectar los vectores de observación, así es posible utilizar los vectores de pesos de esta transformación como características que pueden ser clasificadas por algoritmos típicos.

Sea $\xi_{ij}(t)$ la variable dinámica j perteneciente a la observación i, donde $i = 1, 2, \ldots, n$ es el número de observación, $j = 1, 2, \ldots, p$ es el número de variables por observación y $t = 1, 2, \ldots, T$ la longitud de la variable dinámica. Entonces, la matriz de datos correspondiente a una sola observación de tamaño $(T \times p)$ está dada por:

$$\mathbf{X}_i = \begin{bmatrix} \xi_{i1}(1) & \xi_{i2}(1) & \cdots & \xi_{ip}(1) \\ \xi_{i1}(2) & \xi_{i2}(2) & \cdots & \xi_{ip}(2) \\ \vdots & \vdots & & \vdots \\ \xi_{i1}(T) & \xi_{i2}(T) & \cdots & \xi_{ip}(T) \end{bmatrix} \tag{C.1}$$

A partir de la matriz definida en (C.1), se construye el correspondiente vector observación Γ_i de tamaño $(pT \times 1)$,

$$\Gamma_i = \begin{bmatrix} \xi_{i1}(1) \\ \xi_{i1}(2) \\ \vdots \\ \xi_{i1}(T) \\ \xi_{i2}(1) \\ \vdots \\ \xi_{i2}(T) \\ \vdots \\ \xi_{ip}(1) \\ \vdots \\ \xi_{ip}(T) \end{bmatrix} \tag{C.2}$$

y la observación promedio del conjunto de observaciones de entrenamiento está definida por,

$$\bar{\Gamma} = \frac{1}{n} \sum_{i=1}^{n} \Gamma_i \tag{C.3}$$

por tanto, cada observación difiere de la observación promedio por el vector,

$$\Phi_i = \Gamma_i - \bar{\Gamma} \tag{C.4}$$

Una vez construidos los vectores Φ_i, se hallan los componentes principales de la distribución de las observaciones, para ello se realiza la técnica PCA, con la cual se busca un conjunto de n vectores ortonormales $\mathbf{v}_i$ y sus valores propios asociados λ_i no nulos, que representan de mejor forma la estructura original de los datos. Del total de valores y vectores propios, pueden escogerse los m mejores, en el sentido de la cantidad de información que representan, por medio de algún criterio desarrollado para tal fin [58]. La matriz de covarianza $\mathbf{S}$, que relaciona los puntos de las variables dinámicas, a partir de la cual se calculan los valores y vectores propios está dada por,

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n} \Phi_i \Phi_i^\top = \frac{1}{n}\mathbf{G}\mathbf{G}^\top \tag{C.5}$$

donde, la matriz $\mathbf{G}$ es,

$$\mathbf{G} = \begin{bmatrix} \Phi_1 & \Phi_2 & \cdots & \Phi_n \end{bmatrix}$$

Calcular la matriz de covarianza $\mathbf{S}$ de tamaño $(pT \times pT)$ y obviamente los pT valores propios y los pT vectores propios de ella, puede ser un proceso computacionalmente costoso. Sin embargo, es posible realizar este procedimiento de forma más eficiente [79]. En vez de determinar los vectores y valores propios a partir de $\mathbf{G}\mathbf{G}^\top$, se considera la matriz $\mathbf{G}^\top\mathbf{G}$ de tamaño $(n \times n)$, para determinar los n valores propios λ y los nuevos n vectores propios $\hat{\mathbf{v}}$; lo anterior se realiza porque usualmente en la práctica $n \ll pT$. La relación que existe entre los vectores propios $\mathbf{v}$ y los vectores propios $\hat{\mathbf{v}}$ puede describirse por,

$$\mathbf{G}^\top\mathbf{G}\hat{\mathbf{v}}_i = \lambda\hat{\mathbf{v}}_i$$

premultiplicando por $\mathbf{G}$,

$$\mathbf{G}\mathbf{G}^\top\mathbf{G}\hat{\mathbf{v}}_i = \lambda\mathbf{G}\hat{\mathbf{v}}_i$$

de donde,

$$\begin{aligned} \mathbf{S}\mathbf{G}\hat{\mathbf{v}}_i &= \lambda\mathbf{G}\hat{\mathbf{v}}_i \\ \mathbf{S}\mathbf{v}_i &= \lambda\mathbf{v}_i \end{aligned}$$

entonces,

$$\mathbf{v}_i = \mathbf{G}\hat{\mathbf{v}}_i \tag{C.6}$$

Por tanto $\mathbf{G}^\top\mathbf{G}$ y $\mathbf{G}\mathbf{G}^\top$ tienen los mismos valores propios y sus vectores propios están relacionados por (C.6). Mientras que de $\mathbf{G}\mathbf{G}^\top$ se pueden tener pT valores propios, de $\mathbf{G}^\top\mathbf{G}$ sólo se pueden tener n valores propios. Sin embargo, estos n valores propios corresponden con a los n valores propios mayores de $\mathbf{G}\mathbf{G}^\top$, es importante tener en cuenta que los vectores propios $\mathbf{v}_i$ deben normalizarse, tal que, $\|\mathbf{v}_i\| = 1$.

Una vez se han calculado n valores propios y n vectores propios, es posible seleccionar sólo m vectores propios (donde $m < n$), asociados a los m mayores valores propios, por medio de algún criterio desarrollado para tal fin [58]. Por otra parte, debido a que PCA es una transformación lineal de los datos, es posible reconstruir una observación a partir de la suma ponderada de los vectores propios, por medio de,

$$\hat{\Phi}_i = \sum_{k=1}^{m} w_k \mathbf{v}_k \tag{C.7}$$

donde, $\hat{\Phi}_i$ es la observación reconstruida y los pesos de ponderación están dados por,

$$w_k = \mathbf{v}_k^\top \Phi_i \tag{C.8}$$

En este sentido, cada observación normalizada está representada en la base, por un vector de pesos $\mathbf{\Omega}$, donde,

$$\mathbf{\Omega}_i = \begin{bmatrix} w_{i1} & w_{i2} & \cdots & w_{im} \end{bmatrix} \tag{C.9}$$

y de esta forma, los pesos son las coordenadas de las observaciones de entrenamiento, que constituyen las clases (patrones) en un nuevo espacio dado por los $\mathbf{v}_k,\ k = 1, \ldots, m$ vectores propios calculados.
Para aplicar el procedimiento descrito debe tenerse en cuenta que:

- Exista independencia estadística entre observaciones.
- La cantidad de observaciones empleadas deben ser suficientes para que el experimento tenga significancia estadística.
- Se asume sincronía en la secuencia de ventanas que se analizan en la variables dinámicas, es decir que, cada observación se considera como una función muestra del mismo proceso aleatorio. En este sentido, la matriz de covarianza que se analiza está constituida por promedios de ensamble (*ensemble averages*).

Finalmente, para validar una nueva observación Γ_{val}, el proceso consiste en:

1. Normalizar la muestra, $\Phi_{val} = \Gamma_{val} - \bar{\Gamma}$
2. Proyectar la muestra de validación normalizada en el espacio de vectores propios $\mathbf{v}_k$.
3. Representar la muestra de validación en el espacio de los vectores propios, por medio su vector pesos $\mathbf{\Omega}_{val}$ y emplear un algoritmo de clasificación para determinar la clase a la que corresponde dicha muestra.

Por otra parte, la técnica PCA no sólo realiza la extracción de características, sino que también permite la identificación y selección de las variables que de mayor forma contribuyen al proceso de reconocimiento. Esta identificación se logra analizando primero el vector $\boldsymbol{\rho}$ de tamaño $(pT \times 1)$

$$\boldsymbol{\rho} = \sum_{k=1}^{m} |\lambda_k \mathbf{v}_k| \tag{C.10}$$

los mayores valores dentro del vector $\boldsymbol{\rho}$ señalan cada uno de los puntos de las variables dinámicas que más influencia tienen en el proceso. Luego, si se reordena el vector $\boldsymbol{\rho}$ en

forma de la matriz $\mathbf{P}$, tal que,

$$\boldsymbol{\rho} = \begin{bmatrix} \rho_{11} \\ \rho_{12} \\ \vdots \\ \rho_{1T} \\ \rho_{21} \\ \vdots \\ \rho_{2T} \\ \vdots \\ \rho_{p1} \\ \vdots \\ \rho_{pT} \end{bmatrix} \Rightarrow \mathbf{P} = \begin{bmatrix} \rho_{11} & \rho_{21} & \cdots & \rho_{p1} \\ \rho_{12} & \rho_{22} & \cdots & \rho_{p2} \\ \vdots & \vdots & & \vdots \\ \rho_{1T} & \rho_{2T} & \cdots & \rho_{pT} \end{bmatrix}$$

se identifican las variables dinámicas de más influencia en el proceso, como aquellas para las cuales, los valores $\hat{\rho}_j$ sean mayores, siendo,

$$\hat{\rho}_j = \sum_{t=1}^{T} \rho_{jt},\ j = 1, \ldots, p \tag{C.11}$$

porque estas características son las que tienen más alta correlación con los componentes principales.

Bibliografía

[1] X. Wang and K. K. Paliwal, "Feature extraction and dimensionality reduction algorithms and their applications in vowel recognition," *Pattern Recognition*, vol. 36, pp. 2429 – 2439, 2003.

[2] A. K. Jain, R. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 22, no. 1, pp. 4–37, January 2000.

[3] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. John Wiley & Sons, INC, 2001.

[4] M. Álvarez, R. Henao, G. Castellanos, J. Godino-Llorente, and A. Orozco, "Kernel principal component analysis through time for voice disorder classification," in *Proceedings of The 28th International Conference of the IEEE Engineering in Medicine and Biology Society*, New York, USA, August 2006.

[5] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Transactions on Signal Processing*, vol. 40, no. 12, pp. 3043–3053, 1992.

[6] M. Wester, "Automatic classification of voice quality: Comparing regression models and hidden markov models," *Proceedings of VOICEDATA98, Symposium on Databases in Voice Quality Research and Education*, 1998.

[7] J. D. Arias-Londoño, M. Álvarez, G. Castellanos-Domínguez, and J. I. Godino-Llorente, "Caracterización dinámica de señales de ECG con infarto agudo de miocardio usando HMM," *Congreso anual de la sociedad española de ingeniería biomédica - CASEIB*, 2005.

[8] A. Dibazar and S.Ñarayanan, "A system for automatic detection of pathological speech," in *Proceedings of the 36th Asilomar Conf. Signals, Systems & Computers.*

[9] M. Palacios, M. Vallverdú, D. Hoyer, F. Clarià, R. Baranowski, and P. Caminal, "Análisis de la Dinámica de la VRC mediante Modelos Ocultos de Markov: Pacientes con cardiomiopatia hipertrófica," Tech. Rep., 2004.

[10] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 3, pp. 257–265, 1997.

[11] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of The IEEE*, vol. 77(2), February 1989.

[12] P. Micó, "Nuevos desarrollos y aplicaciones basados en métodos estocásticos para el agrupamiento no supervisado de latidos en señales electrocardiográficas," Ph.D. dissertation, Departamento De Informática De Sistemas Y Computadores, Universidad Politécnica De Valencia, Diciembre de 2005.

[13] M. Collins, "The EM algorithm," University of Pennsylvania, Tech. Rep., 1997.

[14] J. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," International Computer Science Institute, Berkeley CA, USA. 94704, Tech. Rep., 1998.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[16] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, pp. 47–60, 1996.

[17] S. Borman, "The expectation maximization algorithm a short tutorial," Tech. Rep., Last updated June 28, 2006. [Online]. Available: http://www.seanborman.com/publications/EM_algorithm.pdf#search=%22generalized%20expectation%20maximization%22

[18] M. A. Álvarez, "Reconocimiento de Voz Sobre Diccionarios Reducidos usando Modelos Ocultos de Markov," Tesis pregrado, 2004.

[19] V. Valtchev, J. Odell, P. Woodland, and S. Young, "MMIE training of large vocabulary recognition systems," *Speech Communication*, vol. 22, pp. 303–314, 1997.

[20] W. Reichl and G. Ruske, "Discriminative training for continuous speech recognition," Institute for Human-Machine-Comunication, Munich University of Technology, München, Germany, Tech. Rep., 1996.

[21] S. Riis, "Hidden markov models and neural networks for speech recognition," Ph.D. dissertation, Technical University of Denmark, 1998. [Online]. Available: citeseer.ist.psu.edu/riis98hidden.html

[22] P. S. Gopalakrishnan, D. Kanevsky, N. Arthur, and D.Ñahamoo, "An inequality for rational functions with applications to some statistical estimation problems," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 107–113, 1991.

[23] B.-Y. Assaf and D. Burshtein, "A discriminative training algorithm for hidden markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 3, pp. 204–217, May 2004.

[24] Y.Ñormandin, R. Cardin, and R. De-Mori, "High-performance conected digit recognition using maximum mutual information estimation," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 299–311, 1994.

[25] D. A. Reynolds, "Speaker identification and verification using gaussian mixture speaker models," *Speech Comunications*, vol. 17, pp. 91–108, 1995.

[26] A. Biem, "Minimum classification error training for online handwrite recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1041–1051, 2006.

[27] R. B. Lyngsø, C. Pedersen, and H.Ñielsen, "Measures on Hidden Markov Models," Basic Research in Computer Science BRICS, Tech. Rep. 99-6, Feb. 1999.

[28] L. Gavidia-Ceballos and J. Hansen, "Direct Speech Feature Estimation Using an Iterative EM Algorithm for Vocal Fold Pathology Detection," *IEEE Transactions on Biomedical Engineering*, vol. 43, no. 4, pp. 373–383, April. 1996.

[29] A. Dibazar and S.Ñarayanan, "A System for Automatic Detection of Pathological Speech," in *Proceedings of the 36th Asilomar Conf. Signals, Systems & Computers*, 2002.

[30] A.Ñogueiras, A. Moreno, A. Bonafonte, and J. Mariño, "Speech Emotion Recognition Using Hidden Markov Models," in *Proceedings of Eurospeech*, Scandinavia, 2001.

[31] M. C. Nechyba and Y. Xu, "Stochastic similarity for validating human control strategy models," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 3, pp. 437–451, 1998.

[32] D. Gao, M. K. Reiter, and D. Song, "Behavioral Distance Measurement Using Hidden Markov Models," in *Procedings of LNCS 4219*. Berlin Heidelberg: Springer-Verlag, 2006, pp. 19–40.

[33] M. Falkhausen, H. Reininger, and D. Wolf, "Calculation of Distance Measures Between Hidden Markov Models," Tech. Rep.

[34] B.-H. Juang and L. Rabiner, "A Probabilistic Distance Measure for Hidden Markov Models," *AT&T Technical Journal*, vol. 64, no. 2, pp. 391–408, Feb. 1985.

[35] R. Dugad and U. B. Desai, "A Tutorial on Hidden Markov Models," Signal Processing and Artifial Neural Networks Laboratory, Indian Institute of Technology - Bombay, Tech. Rep. SPANN-96.1, May 1996.

[36] M. Do, "Fast approximation of kullback-leibler distance for dependence trees and hidden markov models," *IEEE Signal Processing Letters*, vol. 10, no. 3, pp. 115–118, 2003.

[37] T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[38] L. Xie and V. A. Ugrinovskii, "A Posteriori Probability Distances Between Finite-Alphabet Hidden Markov Models," *IEEE Transactions on Information Theory*, vol. 53, no. 2, pp. 783–793, Feb. 2007.

[39] X. Wang, "Feature extraction and dimensionality reduction in pattern recognition and their application in speech recognition," Ph.D. dissertation, School of Microelectronic Engineering, Faculty of Engineering and Information Technology, Griffith University, Nov. 2002.

[40] D. Peña, *Análisis de datos multivariantes.* Mc Graw Hill, 2002.

[41] W. Krzanowski, "Principal component analysis in the presence of group structure," *Applied Statistics*, vol. 33, pp. 164 – 168, 1984.

[42] T. Li, S. Zhu, and M. Ogihara, "Using discriminant analysis for multi-class classification," in *Third IEEE International Conference on Data Mining*, 2003.

[43] M. Loog, R. Duin, and R. Haeb-Umbach, "Multiclass linear dimension reduction by weighted pairwise fisher criteria," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 23, no. 7, p. 762–766, 2001.

[44] X. Wang and K. K. Paliwal, "A modified minimum classification error (mce) training algoritm for dimensionalitu reduction," *Journal of VLSI Signal Processing*, vol. 32, pp. 19–28, 2002.

[45] A. Jennings and J. J. McKeown, *Matrix Computation*, 2nd ed. John Wiley & Sons, 1992.

[46] E. Kreyszig, *Introductory funtional analysis with applications.* John Wiley & Sons, 1978.

[47] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, "Feature selection for svms," in *Neural Information Processing Systems*, 2001.

[48] Massachusetts Eye and Ear Infirmary, "Voice disorders database. versión 1.03," [CD-ROM],Lincoln Park, NJ: Kay Elemetrics Corp, 1994.

[49] V. Parsa and D. Jamieson, "Identification of pathological voices using glottal noise measures," *Journal of Speech Language and Hearing Research.*, vol. 43, no. 2, pp. 469–485, Apr. 2000.

[50] J. R. Deller, J. G. Proakis, and J. H. Hansen, *Discrete-Time Processing of Speech Signals*, J. Griffin, Ed. Macmillan Publishing Company, 1993.

[51] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition.* PTR Prentice Hall, 1993.

[52] J. I. Godino-Llorente, P. Gómez-Vilda, N. Sáenz-Lechón, M. Blanco-Velasco, F. Cruz-Roldán, and M. A. Ferrer-Ballester, "Discriminative methods for the detection of voice disorders." Proceedings of the 3th International Conference on Non-Linear speech processing, Barcelona, Spain, 2005.

[53] D. Deliyski, "Acoustic model and evaluation of pathological voice production," in *Proceedings of Eurospeech '93*, vol. 3, Berlin, Germany, 1993, pp. 1969–1972.

[54] H. Kasuya, S. Ogawa, K. Mashima, and S. Ebihara, "Normalized noise energy as an acoustic measure to evaluate pathologic voice," *Journal of the Acoustical Society of America*, vol. 80, no. 5, pp. 1329–1334, Nov 1986.

[55] D. Michaelis, T. Gramms, and H. W. Strube, "Glottal-to-noise excitation ratio - a new measure for describing pathological voices," *Acustica/Acta acustica*, vol. 83, pp. 700–706, 1997.

[56] J. Godino-Llorente, P. Gómez-Vilda, and M. Blanco-Velasco, "Dimensionality reduction of a pathological voice quality assessment system based on gaussian mixture models and short-term cepstral parameters," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 10, pp. 1943–1953, 2006.

[57] I. T. Jolliffe, *Principal component analysis*, 2nd ed., ser. Springer series in statistics. New York, NY, USA: Springer, 2002.

[58] G. Daza-Santacoloma, "Metodología de reducción de dimensión para sistemas de reconocimiento automático de patrones sobre bioseñales," Master's thesis, Universidad Nacional de Colombia, 2006.

[59] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, Mar. 1998.

[60] C. M. Bishop, *Neural Networks for Pattern Recognition*, 2nd ed. Oxford University Press, 1995.

[61] N. Sáenz-Lechón, J. Godino-Llorente, V. Osma-Ruiz, and P. Gómez-Vilda, "Methodological issues in the development of automatic systems for voice pathology detection," *Biomedical Signal Processing and Control*, vol. 1, no. 2, pp. 120–128, 2006.

[62] T. Fawcett, "ROC graphs: Notes and practical considerations for researchers," HP Laboratories, Palo Alto, CA, Tech. Rep., March 2004.

[63] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, Apr. 1982.

[64] A. F. Martin, G. R. Doddington, T. Kamm, M. Ordowski, and M. A. Przybocki, "The det curve in assessment of detection task performance," in *Proceedings of Eurospeech '97*, vol. IV, Rhodes, Crete,, 1997, pp. 1895–1898.

[65] A. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[66] N. Sáenz-Lechón, "Sistemas de detección automática de patología vocal," Trabajo de investigación para la obtención del Diploma de Estudios Avanzados del programa de doctorado "Tecnologías de la Información y las Comunicaciones", Universidad Politécnica de Madrid, 2005.

[67] S. Cruz-Llanas, "Integración de audio y vídeo en reconocimiento biométricio," Ph.D. dissertation, Escuela Técnica Superior de Ingenieros de Telecomunicacione, Universidad Politécnica de Madrid, España, 2005.

[68] J. I. Godino-Llorente, S. Aguilera-Navarro, and P. Gómez-Vilda, "Automatic detection of voice impairments due to vocal misuse by means of gaussian mixture models," in *Proceedings of IEEE EMBS '01*, vol. 2, Istanbul, Turkey, Oct. 2001, pp. 1723–1726.

[69] J. Ferreiros and J. M. Pardo, "Improving continuous speech recognition in spanish by phone-class semicontinuous HMMs with pausing and multiple pronunciations," *Speech Communication*, vol. 29, no. 1, pp. 65–76, Sept. 1999.

[70] G. R. Doddington, M. A. Przybocki, A. F. Martin, and D. A. Reynolds, "The nist speaker recognition evaluation - overview, methodology, systems, results, perspective," *Speech Communication*, vol. 31, no. 2-3, pp. 225–254, June 2000.

[71] G. T. Toussaint, "Bibliography on estimation of misclassification," *IEEE Transactions on Information Theory*, vol. 2, no. 4, pp. 472–479, July 1974.

[72] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the International Joint Conference on Artificial Intelligence IJCAI*, 1995.

[73] R. O. Duda and P. E. Hart, *Pattern classification and scene analysis.* New York: John Wiley & Sons, 1973.

[74] D. Peña, *Fundamentos de estadística,.* Madrid: Alianza Editorial, 2001.

[75] W. G. Baxt and H. White, "Bootstrapping confidence intervals for clinical input variable effects in a network trained to identify the presence of acute myocardial infarction," *Neural Computation*, vol. 7, pp. 624–638, 1995.

[76] S. Haykin, *Neural networks.* New York: Macmillan, 1994.

[77] J. A. Hanley and B. J. McNeil, "A method of comparing the areas under receiver operating characteristics curves derived from the same cases," *Radiology*, vol. 148, no. 3, pp. 839–843, Sept. 1983.

[78] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 1991, pp. 586–591.

[79] ——, "Eigenfaces for recognition," *Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.