UNIVERSIDAD NACIONAL DEL LITORAL
Facultad de Ingeniería y Ciencias Hídricas

Información Discriminativa en Clasificadores Basados en Modelos Ocultos de Markov

Diego Tomassi

Tesis remitida al Comité Académico del Doctorado
como parte de los requisitos para la obtención del grado de
DOCTOR EN INGENIERIA
Mención Inteligencia Computacional, Señales y Sistemas de la
UNIVERSIDAD NACIONAL DEL LITORAL

2010

Comisión de Posgrado, Facultad de Ingeniería y Ciencias Hídricas,
Ciudad Universitaria, Paraje “El Pozo”, S3000, Santa Fe, Argentina.

Doctorado en Ingeniería
Mención Inteligencia Computacional, Señales y Sistemas

Título de la obra:

Información Discriminativa en Clasificadores Basados en Modelos Ocultos de Markov

Autor: Diego Tomassi
Director: Dr. Diego Milone
Codirector: Dra. Liliana Forzani

Lugar: Santa Fe, Argentina

Palabras Claves:

Modelos ocultos de Markov
Aprendizaje discriminativo
Error de clasificación mínimo
Reducción de dimensiones
Análisis discriminante

Resumen en Español

En la actualidad, una cantidad enorme de información se registra y almacena diariamente en forma de imágenes, video, audio, señales biomédicas, datos financieros y científicos. Para sacar provecho de toda esta información, es útil encontrar regularidades y estructuras en los datos que permitan reconocer patrones y clasificarlos de forma conveniente. La automatización de ese proceso es el objeto del aprendizaje maquinal.

En aplicaciones como el reconocimiento de la escritura manuscrita, del habla o de objetos en grabaciones de video, las entidades que se desean clasificar se presentan como una sucesión o secuencia de datos correlacionados entre sí y la asignación de cada secuencia a una clase determinada se basa en el modelado estadístico de las mismas. Es posible considerar que secuencias distintas son independientes, pero es necesario describir adecuadamente las dependencias estadísticas entre las observaciones que las constituyen. Los modelos ocultos de Markov (HMM, del Inglés Hidden Markov Model) son la herramienta más utilizada con este propósito. El atractivo principal de estos modelos reside en su simpleza, en la disponibilidad de algoritmos muy eficientes desde el punto de vista computacional para su entrenamiento y evaluación, y en su capacidad para describir secuencias con un número variable de observaciones.

En un escenario de clasificación típico, los datos observados pertenecen a una de h clases distintas, pero puede usarse un mismo conjunto de características para describir a todas las clases. Si $Y = 1, 2, \ldots, h$ denota la clase y $\mathbf{X} \in \mathbb{R}^p$ las características, el clasificador es una función $f(\mathbf{X})$ que nos indica la clase a la cual pertenece $\mathbf{X}$ con una mínima probabilidad de error.

El reconocimiento estadístico de patrones comprende fundamentalmente la selección de características útiles para discriminar entre las distintas clases, el modelado estadístico de las mismas, y la construcción de $f(\mathbf{X})$ a partir de tales modelos. En los problemas que nos interesan en esta tesis, los datos que queremos clasificar son secuencias de observaciones $\mathbf{X} = \{X_1, X_2, \ldots, X_{T_i}\}$ y la regla de clasificación $f(\mathbf{X})$ utiliza un HMM ϑ_y asociado con los datos de cada clase y.

Tradicionalmente, el uso de HMM para construir un clasificador se encuadra dentro de las estrategias generativas de aprendizaje automático. Bajo este enfoque, la suposición fundamental es que los datos de cada clase son modelados exactamente por el HMM correspondiente, de modo que $\mathrm{p}(\mathbf{X}|Y = y) \sim \mathrm{p}(\mathbf{X}|\vartheta_y)$. Suponiendo que se conocen también las probabilidades *a priori* $\mathrm{p}(Y = y)$, el clasificador óptimo es la regla de Bayes, que asigna a los datos una clase de acuerdo al modelo que maximiza la probabilidad posterior $\mathrm{p}(\vartheta_y|\mathbf{X})$. El aprendizaje del clasificador se reduce entonces a estimar las distribuciones $\mathrm{p}(\mathbf{X}|\vartheta_y)$ a partir de un conjunto de datos de entrenamiento, para lo cual se usa comúnmente estimación de máxima verosimilitud.

Entrenando los clasificadores de esta forma se han logrado buenos desempeños en aplicaciones que involucran, por ejemplo, la clasificación de la escritura manuscrita, del habla y de secuencias biológicas como proteinas y ácidos nucleicos. No obstante, en este enfoque se tratan los datos de cada clase en foma independiente y no se aprovecha todo el conjunto de datos disponibles para enfatizar las diferencias entre las distintas clases. El objetivo general de esta tesis es proveer nuevas herramientas para construir clasificadores de datos secuenciales basados en HMM que aprovechen mejor la información disponible para ayudar a discriminar entre las clases.

Aprendizaje discriminativo de HMM definidos en el dominio de la transformada ondita

Una observación clave respecto del aprendizaje generativo es reconocer que la densidad $\mathrm{p}(\mathbf{X}|\vartheta_y)$ usada en un clasificador no es idéntica a la verdadera distribución de los datos de la clase, sino que usualmente es sólo una aproximación escogida por su conveniencia analítica y computacional. En consecuencia, el clasificador de Bayes basado en $\mathrm{p}(\mathbf{X}|\vartheta_y)$ comúnmente deja de ser óptimo en aplicaciones prácticas.

Debido a ello, en los últimos años se ha registrado un gran interés por el entrenamiento discriminativo de HMM. A diferencia del entrenamiento convencional, en este tipo de aprendizaje el objetivo ya no es describir adecuadamente $\mathrm{p}(\mathbf{X}|Y = y)$, sino construir directamente una función $f(\mathbf{X}; \vartheta_1, \ldots, \vartheta_h)$ que minimice la tasa de error esperada en la clasificación. Para ello, los parámetros de todos los modelos se estiman simultáneamente, utilizando datos de entrenamiento de todas las clases. Una alternativa directa para optimizar el desempeño del clasificador es minimizar el *riesgo empírico* de clasificación con respecto a una función de costo. La elección usual para esta función es asignar un costo nulo cuando la clase asignada a la observación es correcta y un costo unitario en cualquier otro caso.

La estimación de HMM con este tipo de técnicas ha mostrado resultados muy interesantes en diversas aplicaciones. Sin embargo, estos algoritmos están desarrollados para entrenar HMM con una estructura particular en la cual la distribución condicional de las observaciones es una densidad normal o una mezcla de densidades normales. Aunque este tipo de HMM es el usado con mayor frecuencia en las aplicaciones, no resulta adecuado para describir algunas secuencias de datos con estructuras de dependencias particulares. Un ejemplo de ello son las representaciones de señales basadas en onditas.

La transformada ondita ha resultado ser una herramienta muy útil para analizar señales e imágenes en distintas aplicaciones, permitiendo su descomposición en elementos con distintos niveles de detalle o resolución. Las representaciones suelen concentrar la energía de toda la señal en un número reducido de coeficientes y aquellos que están relacionados temporal/espacialmente suelen mostrar fuertes dependencias estadísticas a lo largo de las distintas escalas de análisis. El uso de mezclas de densidades normales definidas sobre el conjunto de coeficientes resulta inadecuado para modelar estas propiedades. Por el contrario, un modelo oculto de Markov definido sobre los coeficientes de la transformación ha resultado ser un modelo especialmente útil para estas representaciones. Estos modelos reciben el nombre de árboles ocultos de Markov (HMT, del Inglés Hidden Markov Trees) y se han aplicado con éxito en tareas diversas. Los HMT fueron luego empleados como modelos de observación en HMM convencionales. Esto permitió combinar las ventajas del HMT para capturar dependencias estadísticas locales en el dominio de la transfomación con la capacidad del HMM de modelar relaciones de más largo alcance a lo largo de la secuencia y de tratar con la longitud variable que suelen mostrar las mismas. No obstante, en estos trabajos se estiman los parámetros de los modelos HMM-HMT intentando aproximar la distribución de $p(\mathbf{X}|Y = y)$, sin explotar infomación discriminativa.

En esta tesis, se propone un método discriminativo de estimación de parámetros para modelos compuestos HMM-HMT con el objeto de mejorar su desempeño en tareas de clasificación. La estrategia desarrollada utiliza un conjunto de funciones discriminantes, definidas a partir de la máxima probabilidad que pueden presentar los datos observados bajo el modelo HMM-HMT correspondiente a cada clase. Estas cantidades pueden ser computadas en forma eficiente utilizando una adaptación del algoritmo de Viterbi. Partiendo de modelos parcialmente entrenados bajo el enfoque de máxima verosimilitud, el método adapta iterativamente los parámetros del conjunto de modelos a fin de minimizar una aproximación diferenciable del riesgo de la clasificación sobre el conjunto de datos de entrenamiento. El aprendizaje es supervisado y la aproximación de la función de riesgo se construye en tres pasos:

- Las funciones discriminantes se combinan en una única medida $d(\mathbf{X})$ cuyo signo decide si la clase asignada a la secuencia de entrenamiento $\mathbf{X}$ es correcta: $f(\mathbf{X}) = \text{sign}[d(\mathbf{X})]$ y la clasificación es correcta si $f(\mathbf{X}) < 0$.

- Una función de costo asociada a la clasificación de $\mathbf{X}$ penaliza una decisión equivocada. $\ell(d)$ es una función continua que se aplica sobre el rango de valores de $d(\mathbf{X})$ para otorgar un valor en el intervalo $[0; 1]$. Al ser una función continua de d, este costo puede penalizar no sólo la decisión final del clasificador sino también la dificultad que presenta esa decisión, ya que valores $|d(\mathbf{X})|$ cercanos a cero indican que la secuencia $\mathbf{X}$ presenta una probabilidad similar de pertenecer a clases distintas.

- La función de riesgo es la suma de los costos asociados con la clasificación de todas las secuencias de entrenamiento.

El riesgo resultante es una función de los parámetros de los modelos a través de las funciones discriminantes que se combinan en d. Decimos que es aproximada porque no utiliza la función de costo 0-1, que es discontinua, sino una aproximación diferenciable dada por $\ell(d)$. Esto nos permite obtener su gradiente con respecto al conjunto de parámetros de los modelos y de esa forma utilizar métodos de gradiente para hallar los estimadores que minimizan la función de riesgo.

Proponemos y comparamos dos alternativas para la selección de $d(\mathbf{X})$. Ambas comparan el valor de la función discriminante correspondiente a la clase correcta de $\mathbf{X}$ con una aproximación suave al máximo valor que toman las funciones discriminantes para el resto de las clases, de modo de ver qué tan difícil de clasificar resulta $\mathbf{X}$. Sin embargo, una de las alternativas efectúa esta comparación a través de una diferencia mientras que la otra alternativa lo hace a través de un cociente que se compara luego con la unidad. En el primer caso, $d \in (-\infty; +\infty)$ y el gradiente de la función de costo usado en la actualización de los parámetros resulta ser una función de $|d(\mathbf{X})|$. Es decir que el aprendizaje está conducido por la dificultad que presentan las secuencias de entrenamiento para ser clasificadas correctamente, independientemente de que la decisión del clasificador resulte correcta o no. De esta forma, una secuencia que es clasificada correctamente con facilidad no genera una modificación apreciable en el valor de los parámetros. Tampoco lo hace una secuencia que es clasificada incorrectamente presentando un valor positivo muy grande de $d(\mathbf{X})$. Por el contrario, para la segunda definición de $d(\mathbf{X})$ el rango de esta función es $(-\infty; 1]$ y entonces las secuencias para las cuales el clasificador se equivoca fuertemente presentan un $d(\mathbf{X})$ cercano a la unidad. Para una misma función de costo $\ell(d)$ que es simétrica en d, la consecuencia de esto es que los datos de entrenamiento que son mal clasificados durante el aprendizaje del clasificador provocan actualizaciones de los parámetros que en general son de mayor magnitud que las registradas con la alternativa anterior, de modo que las secuencias mal clasificadas tienen más peso sobre el proceso de aprendizaje.

Para evaluar ambas alternativas se realizaron pruebas de reconocimiento de fonemas extraídos de la base de datos TIMIT, de referencia en aplicaciones de reconocimiento automático del habla. Los fonemas escogidos representan una prueba de gran dificultad para un clasificador, ya que estas señales están obtenidas de registros de habla continua, lo que suma a las semejanzas acústicas una gran variabilidad de los fonemas debida al contexto en el que fueron enunciados. En ambos casos, los resultados obtenidos mostraron ser consistentemente mejores que los obtenidos con clasificadores basados en modelos entrenados de forma tradicional. No obstante, las mejoras de desempeño registradas fueron significativamente mayores para la segunda alternativa, que penaliza con mayor intensidad los casos que son mal clasificados. En estos ejemplos, las tasas de error de clasificación mostraron reducciones cercanas al 20% comparadas con las correspondientes a clasificadores entrenados por los métodos tradicionales.

Los resultados correspondientes a esta parte del trabajo de tesis fueron publicados en [88, 89]. Durante la primera parte de estos desarrollos, se exploró también el uso de los HMM-HMT para aplicaciones de supresión de ruido basada en modelos estadísticos. Esos primeros resultados fueron reportados en [72].

Reducción de dimensiones bajo el enfoque de suficiencia

Cuando se usan modelos estadísticos para el reconocimiento de patrones, es frecuente incluir un procedimiento para reducir la dimensión p del espacio de características. Ello permite definir modelos con un menor número de parámetros, de modo que fijado el conjunto de datos de entrenamiento, la varianza de los estimadores obtenidos es menor que si se hubieran definido modelos más grandes sobre las características originales. Esta disminución de la varianza de los estimadores usualmente se traduce en una mejora en el desempeño del clasificador.

En los métodos lineales de reducción de dimensiones las características originales se proyectan a un subespacio de menor dimensión mediante una transformación lineal. En el contexto de clasificadores basados en modelos ocultos de Markov, los métodos más usados en las aplicaciones son extensiones del análisis discriminante lineal (LDA, del Inglés Linear Discriminant Analysis) para datos normalmente distribuidos. Estos métodos están adaptados a un esquema de estimación de máxima verosimilitud a fin de poder integrar la reducción de dimensiones al proceso tradicional de estimación de parámetros en HMM. La más usada de estas técnicas es una variante conocida simplemente como HLDA (por Heteroscedastic Linear Discriminant Analysis).

Este proceso de reducción no debería perder información relevante para la clasificación, sino conservar toda la información discriminativa presente en las características originales pero en un número menor de combinaciones lineales de las mismas. Sin embargo, a pesar del uso extendido de HLDA en aplicaciones de reconocimiento de patrones basado en modelos ocultos de Markov, su desarrollo no tiene en cuenta la retención de información y tampoco existe hasta el momento un análisis de su optimalidad en tal sentido.

Por el contrario, la reducción suficiente de dimensiones (SDR, del Inglés Sufficient Dimension Reduction) es un enfoque relativamente reciente que tiene en cuenta explícitamente la pérdida de información. El objetivo de esta metodología es estimar el subespacio generado por $\boldsymbol{\rho} \in \mathbb{R}^{p \times d}$, con $d \leq p$ mínimo, de modo que la distribución condicional de $\mathbf{X}|(\boldsymbol{\rho}^T\mathbf{X}, Y)$ es asintóticamente equivalente a la de

$\mathbf{X}|\boldsymbol{\rho}^T\mathbf{X}$. Esta condición asegura que la proyección de $\mathbf{X}$ conserva toda la información disponible sobre Y. Cuando se dispone de un modelo para $\mathbf{X}|(Y = y)$, la estimación de ese subespacio mínimo puede efectuarse usando máxima verosimilitud. Los métodos disponibles de SDR basados en este tipo de estimación se limitan, sin embargo, a datos con distribución normal y han estado orientados típicamente al problema de regresión más que a la clasificación.

En clasificación, el objetivo de la reducción suficiente es estimar el subespacio generado por $\boldsymbol{\rho}$ de modo que $f(\boldsymbol{\rho}^T\mathbf{X}) = f(\mathbf{X})$ para todos los $\mathbf{X}$. Aunque puede parecer que el subespacio estimado de esta forma es distinto al obtenido con la condición anterior, es posible demostrar que cuando los datos de cada clase se distribuyen normalmente ambos subespacios son idénticos.

Partiendo de este resultado, en esta tesis utilizamos desarrollos teóricos recientes referidos a la reducción suficiente de poblaciones normalmente distribuidas para analizar LDA y HLDA en el contexto de suficiencia. Mostramos que las proyecciones obtenidas con LDA conservan la información discriminativa sólo cuando los datos de cada clase se distribuyen normalmente y la matriz de covarianza es la misma para todas las clases. Por otra parte, mostramos que con HLDA es posible lograr una reducción que conserve la información discriminativa, pero que para ello frecuentemente es necesario retener un número grande de combinaciones lineales de las características originales. Esta cantidad usualmente es mayor que la que sería necesario retener empleando otro método de proyección lineal conocido como LAD (por Likelihood Acquired Directions).

Mostramos que este resultado es una consecuencia de la estructura de las matrices de covarianza que implícitamente se suponen en HLDA. La reducción de dimensiones a través de este método puede entenderse como un proceso de dos pasos. En primer lugar, se busca una transformación $(\boldsymbol{\rho}, \boldsymbol{\rho}_0) \in \mathbb{R}^{d\times d}$ de tal modo que toda la información específica de la clase queda concentrada en $\boldsymbol{\rho}^T\mathbf{X}$ y $\boldsymbol{\rho}^T\mathbf{X}$ es estadísticamenete independiente de $\boldsymbol{\rho}_0^T\mathbf{X}$. Luego, como $\boldsymbol{\rho}_0^T\mathbf{X}$ no depende de la clase Y, es común para todas ellas y puede descartarse. La observación fundamental que enfatizamos en esta tesis es que la suposición de independencia entre $\boldsymbol{\rho}^T\mathbf{X}$ y $\boldsymbol{\rho}_0^T\mathbf{X}$ es más fuerte de lo necesario para poder descartar $\boldsymbol{\rho}_0^T\mathbf{X}$ e impone una estructura particular en las matrices de covarianza de los modelos de las clases para poder lograrla. Mostramos que la condición suficiente para reducir las dimensiones sin perder información discriminativa es que $\boldsymbol{\rho}_0^T\mathbf{X}|(\boldsymbol{\rho}^T\mathbf{X}, Y = y)$ no dependa de la clase y. Esta característica es lo que explota LAD y gracias a ello asegura conseguir la reducción suficiente mínima para modelos normales con matrices de covarianza arbitrarias. La consecuencia práctica de estos resultados es que con HLDA usualmente es necesario retener un mayor número de combinaciones lineales de $\mathbf{X}$ que con LAD, o presentado de otra forma, que fijada una cantidad d de combinaciones lineales de las $\mathbf{X}$ originales, estas nuevas coordenadas conservan mejor la información original cuando se obtienen con LAD. Dado que LAD tiene el mismo costo computacional que HLDA, estos resultados sugieren el uso de LAD como alternativa general de reducción lineal para modelos normales con covarianza arbitraria.

Por otra parte, si los datos verdaderamente satisfacen la estructura de covarianza supuesta por HLDA, es posible que la proyección obtenida con este método tampoco sea mínima. En la tesis también proponemos un método de proyección para estos casos que provee una reducción suficiente mínima, al mismo tiempo que explota la estructura particular de las matrices de covarianza. El estimador resultante puede entenderse como una aplicación particular de LAD sobre características transformadas previamente mediante HLDA.

Todos estos desarrollos son ilustrados con simulaciones y con un ejemplo de clasificación de dígitos manuscritos. En este último caso utilizamos HLDA y LAD para proyectar los datos originales a un subespacio bidimensional. El ejemplo ilustra cómo las distintas clases presentan distribuciones de características más normales cuando la reducción se lleva a cabo por medio de LAD. Más importante aún, clasificando los dígitos utilizando un discriminante cuadrático sobre las proyecciones obtenidas con LAD y con HLDA, la tasa de errores de clasificación obtenida con LAD presenta una mejora de aproximadamente el 60% respecto a la tasa de error obtenida con HLDA.

El enfoque de suficiencia para la reducción de dimensiones proporciona además un sustento teórico para inferir cuál debe ser la dimensión d del subespacio al cual se proyectan los datos a fin de conservar toda la información. Este aspecto también es de interés práctico, ya que brinda la posibilidad de utilizar métodos de inferencia menos costosos computacionalmente que las pruebas de validación cruzada utilizadas comúnmente. En la tesis derivamos métodos de inferencia para d usando el criterio de información de Akaike (AIC), el criterio de información de Bayes (BIC), tests de relaciones de verosimilitud (LRT) y tests de permutación. Estos métodos ya estaban disponibles para LAD, pero no así para LDA y HLDA. Las pruebas con datos simulados mostraron que BIC en particular es una buena alternativa para la estimación de d, brindando buenos resultados con un costo computacional relativamente bajo. La opción de menor costo computacional es LRT, pero su desempeño no es tan bueno como el de BIC cuando la cantidad de datos disponibles para el entrenamiento es reducida.

Por último, extendemos todos estos métodos desarrollados inicialmente para datos con distribución normal a HMM que usan densidades normales como modelos de observación. Esta extensión se basa en la descomposición conveniente de la función de verosimilitud que resulta de utilizar el algoritmo de maximización de la esperanza para la estimación de parámetros de los HMM bajo el enfoque de máxima verosimilitud.

Los resultados correspondientes a esta parte del trabajo de tesis fueron publicados en [86]. Por otra parte, el software desarrollado para implementar los métodos de SDR fue publicado en [22].

Contents

List of Figures

Acronyms

AIC	Akaike's information criterion.
BIC	Bayes information criterion.
DWT	Discrete wavelet transform.
EM	Expectation-Maximization.
GHMM	Gaussian hidden Markov model.
GPD	Generalized probabilistic descent.
HLDA	Heteroscedastic linear discriminant analysis.
HMM	Hidden Markov model.
HMT	Hidden Markov tree.
LAD	Likelihood-acquired directions.
LDA	Linear discriminant analysis.
LRT	Likelihood-ratio test.
MAP	Maximum a posteriori.
MCE	Minimum classification error.
MMI	Maximum mutual information.
MLE	Maximum likelihood estimator.
MSE	Mean squared error.
PCA	Principal component analysis.
SDR	Sufficient dimension reduction.
SMF	Symmetric misclassification function.
nSMF	Non-symmetric misclassification function.

Notation

$\mathbf{I}$	Identity matrix.
$\boldsymbol{\rho}$	Basis matrix for a dimension reduction subspace.
$\mathcal{S}_{\boldsymbol{\rho}}$	Subspace spanned by the columns of $\boldsymbol{\rho}$.
$\boldsymbol{\rho}_0$	Basis matrix for the orthogonal complement of $\mathcal{S}_{\boldsymbol{\rho}}$.
$\mathcal{S}_{Y\|\mathbf{X}}$	Central subspace for the regression of Y on $\mathbf{X}$.
$X\|Y$	Random variable X conditioned on the random variable Y.
$X \sim Y$	Asymptotic equivalence of the distributions of X and Y.
$\mathrm{E}_X(X)$	Expectation of the random variable X.
$\mathrm{Var}_X(X)$	Variance of the random variable X.
μ, σ^2	Mean, variance of a scalar random variable.
$\boldsymbol{\mu}$	Mean of a vector random variable.
$\boldsymbol{\mu}_y$	Mean vector of data from population y, $\mathrm{E}(\mathbf{X}\|Y = y)$.
$\boldsymbol{\Sigma}$	Total (marginal) covariance matrix $\mathrm{Var}(\mathbf{X})$.
$\boldsymbol{\Delta}_y$	Conditional (within-class) covariance matrix $\mathrm{Var}(\mathbf{X}\|(Y = y))$.
$\boldsymbol{\Delta}$	Average within-class covariance matrix $\mathrm{E}_Y(\boldsymbol{\Delta}_y)$.
$\mathrm{p}(X)$	Probability density function or probability mass function of X.
$\mathrm{p}(X\|Y = y)$	Conditional pdf of X given the value of Y is y.
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Delta})$	Normal pdf with parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Delta}$.
$\{\cdot\}$	Set or sequence.
$\mathcal{L}_\vartheta$	Likelihood for parametric model ϑ.
$\mathrm{KL}(p\|q)$	Kullback-Leibler divergence between densities p and q.

$\mathbf{X}^n$	n-th observed sequence.
$\mathbf{x}_t^n$	Observed vector at time t in sequence $\mathbf{X}^n$.
$\mathbf{q}_1^t$	Sequence $\{q_1, q_2, \ldots, q_t\}$.
π_k	In a HMM, probability of the chain of being in state k at $t = 1$.
a_{ij}	In a HMM, transition probability from state j to state i.
$b_j(X)$	In a HMM, pdf of observations from state j.
ϑ_i	Hidden Markov model for data from class i.
$\mathcal{T}_u$	In a HMT, subtree rooted at node u.
$\mathcal{T}_{u/v}$	In a HMT, subtree rooted at node u excluding subtree $\mathcal{T}_v$.
$\mathcal{X}_i$	Training sample corresponding to class i.
N_i	Cardinality of $\mathcal{X}_i$.
$\ell(\cdot)$	Loss function.
$\mathcal{R}(\cdot)$	Risk function.
$g_j(\mathbf{X})$	Discriminant function for class j.

CHAPTER 1

Introduction

Learning from data has become a major task in recent years. Collecting and storing data is often easy and cheap with today's technology. However, extracting useful information and taking advantage of it have proved a much more difficult task. Machine learning aims at finding structures in data automatically, so that they can be used as patterns to make predictions about new observations coming from the same source of data.

An important subset of machine learning techniques are targetted to *sequential data*. In this type of data, observations form a correlated sequence. Though different sequences can be assumed independent, modeling the correlations within each of them is fundamental to describe the underlying process. Examples include time series, biomedical signals, handwritten text and sequences of aminoacids in proteins. The observations can come directly from the measurement process, as may be the case with econometric time series, but also from features extracted from a short-term analysis of a whole signal, as it is usual with speech. In addition, the size of the sequences frequently is not fixed, which also contributes to the complexity of modeling them.

Hidden Markov models (HMM) have been found very useful in applications concerning this type of data. They provide parsimonious models of the observations by enabling simple statistical dependencies between latent variables that are hidden but govern the outcomes available to the observer. In a typical setting for classification, the data are assumed to belong to one out of h different classes that can be described using the same set of features or descriptors. Those features are

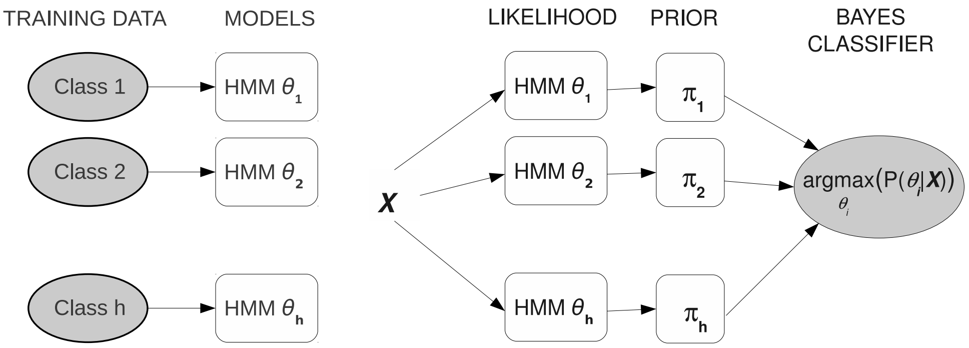


Figure 1.1: *Generative learning approach.*

assumed to be well-modeled by a single HMM for each class. The learning task to build a classifier is to estimate the model parameters that maximizes the likelihood of the class observations given the model. Once all the models have been trained in this way, the classification of a new observation reduces to evaluate which model is more likely to have generated the data.

The learning framework stated above is called *generative*, as it assumes that models can generate the data from their corresponding class. This scheme has shown to be successful for automatic classification tasks concerning for instance speech [28, 51, 80], handwritten characters and digits [5, 9, 47, 36, 90], biological sequences [3], and network traffic [25, 66]. Nevertheless, this basic approach strieves only on describing the data from each class, regardless of whether this effort helps to discriminate beetween classes or not in a practical setting. In this thesis, new learning methods for HMM-based classifiers are developed focussing on discriminative information as a way to improve their performance for pattern recognition.

1.1 Generative vs discriminative learning

Let Y be a label used to indicate the class from where a multivariate vector of features $\mathbf{X}$ comes. Given a sample of labeled observations from the joint process $(Y, \mathbf{X})$, the goal in statistical pattern recognition is to construct a classification rule $Y = f(\mathbf{X})$ to predict with minimum probability of error the class from where an unlabeled vector of features comes. When statistical models are used

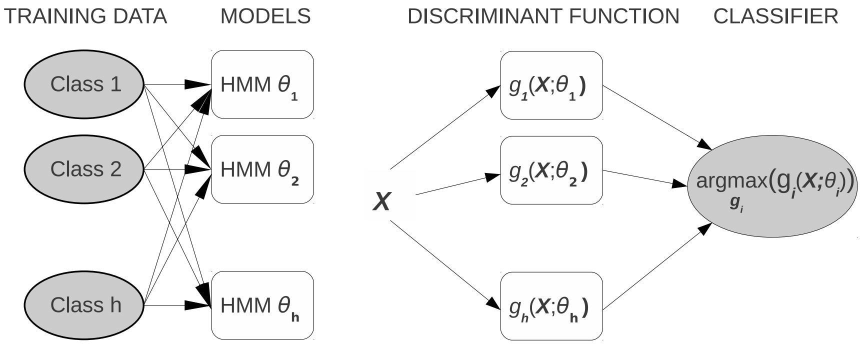


Figure 1.2: *Discriminative learning approach.*

to describe the data, $f(\mathbf{X})$ is a function of those models.

Let $\{\vartheta_y\}$ be the models for the classes, with $y = 1, 2, \ldots, h$. In generative learning, the essential assumption is that $\mathrm{p}(\mathbf{X}|Y = y) = \mathrm{p}(\mathbf{X}|\vartheta_y)$. The exact distribution is not known in advance, but it is common to assume that it belongs to some parametric family of probability density functions, and that the parameters can be estimated from the data. The usual choice to do this is maximum likelihood estimation (MLE). Once all of these distributions and the *a priori* probability of each class $\pi_y = \mathrm{p}(Y = y)$ have been estimated, Bayes rule allows us to compute posterior probabilities for each class given a new observation $\mathbf{X}$. Then, a class label is assigned to $\mathbf{X}$ according to the Bayes classification rule

$$f(\mathbf{X}) = \arg\max_{y} \mathrm{p}(\vartheta_y|\mathbf{X}).$$

This is the usual setting used with HMM-based classifiers. If the assumed models account for the true distribution of the data and the set of training signals is large enough to allow us achive asymptotic optimality of the estimators, the above approach guarantees minimal error rates in classification [75]. Nevertheless, these assumptions hardly ever hold in applications. Assumed models usually cannot be expected to match the true class distributions and sample availability for parameter estimation often is too small to account for the large variability that exists in data. Thus, this approach to classifier design becomes suboptimal and there is a significant increase in error rates [13].

To overcome these limitations, in recent years there has been a growing interest in *discriminative training* of HMMs [13, 45, 52]. Unlike the generative approach, this one does not aim at maximizing the likelihood of the class observations given the model for that class only, but to exploit dissimilarities between models using all the available data. We can think of discriminative learning not trying to describe the whole data distributions, but to locate the decision boundary between them. To do so, this approach uses a set of discriminant functions that depend on the models, and the whole set of parameters $\{\vartheta_y\}$ is estimated simultaneously using training samples from all the classes [52].

Under the discriminative training framework, several criteria have been proposed to drive the learning process of HMM, giving rise to different methods. As examples, maximum mutual information (MMI) [2] seeks to maximize the mutual information between the observations and their labels. This criterion inherits several properties from information theory, but cannot guarantee to achieve the least error rate [13]. On the other hand, minimum classification error (MCE) [54] sets minimization of the error rate explicitely as the optimization task. Minimum phone error (MPE) [79] is another criterion widely known in the speech recognition community. It is conceptually similar to MCE, but when the data is structured at several hierarchical levels it allows to consider smaller units of the sequences to account for the classification error. For example, sentences in speech contain words and words contain phonemes. MCE would account for errors at the sentence level regardless of how many errors occured within the sentence, whereas MPE would account for errors at the phoneme level.

Among these methods, MCE allows for a more direct link between the design of the classifier and its expected performance. Systems trained using this approach have shown important improvements in recognition rates compared to the same systems trained using conventional MLE, both in simple applications [54, 62, 93] as well as in large-scale applications [69, 92]. Nevertheless, up to date these approaches have been limited to HMMs that use Gaussian mixtures as observation distributions.

1.2 Extracting features in the wavelet domain

Observed data are usually transformed in some way before using them for pattern recognition [50]. This process aims to extract features that can help to discriminate better between different classes. Take the speech signal as an example. It is not the rough record what is used for classification, but a number of spectral features obtained in a short-term analysis of the signal [28, 51, 80]. Typically, the speech waveform is analyzed in segments of 30 ms length. For each segment, a spectral analysis is carried out and further processing of the spectrum, gives a set of coefficients that are assumed to be descriptive for the signal. This vector of coefficients is the feature vector used for classification, and statistical models like HMM operate on this feature space. Similar processes for feature extraction could be described for other applications. Most of them are heuristic in nature, specific for the application and they lose information in the process.

Could we think of a feature extraction process that remain fairly the same for a wide range of tasks? One that needs less decisions from an expert and that could be used when smart engineered features are not available in advance? Developing a method like that is obviously a very ambitious goal that would help enormously to automate the learning process. While being far away from a solution yet, first steps in that direction has been given, taking tools from wavelet theory and multirresolution analysis [7, 15, 65].

An important property of the wavelet transform is that it allows to use parsimonious statistical models to describe the coefficients of the representation and the statistical dependencies between them [91]. In this way, useful models are assumed directly on the wavelet domain, and no other feature extraction process is required. The best known of wavelet-domain models is the hidden Markov tree (HMT) [24], which has led to many successful applications [32, 46, 58, 82, 95]. Nevertheless, the HMT is not suitable to sequential data with varying lengths. This limitation arises from the use of the (dyadic) discrete wavelet transform (DWT) [27, 67], which makes the structure of representations depend on the size of signals or images. To overcome this we could think of tying parameters along scales. This is extensively used in the signal processing community, where parameter estimation often relies on a single training sequence. However, in a typical scenario of pattern recognition we have multiple observations available and we would want to use all of that information to train a full model without constraining modeling power because of tying parameters. To do so, the HMT should be trained and used only with signals or images with the same size; otherwise, a warping preprocessing would be required to match different sizes and that would be difficult to achieve in real-time applications.

A different approach to deal with variable length signals in the wavelet domain is to exploit the probabilistic nature of the HMT to embed it as the observation distribution for a standard HMM [71, 73]. In this way, the HMT accounts for local features in a multiresolution framework while the external HMM handles dependencies in a larger time scale and adds flexibility to deal with sequential data. The HMM-HMT model was shown to achieve promising results both for pattern recognition and for denoising tasks [72, 73]. Nevertheless, the training algorithms used so far provide maximum likelihood (ML) estimates of model parameters and discriminative learning approaches have not been proposed yet.

1.3 Dimensionality reduction

The performance of a classifier depends strongly on the set of features on which it acts. As discussed above, observed data are usually transformed in some way to emphasize important information for class discrimination. The output of this feature extraction process is a random vector $\mathbf{X} \in \mathbb{R}^p$ which is assumed to be better suited for classification than the raw measurement.

Nevertheless, the coordinates of $\mathbf{X}$ often have redundant information or some of them are not useful to discriminate between different classes. When this is the case, the parametric models for $\mathbf{X}|(Y = y)$ use parameters to describe nuissance dimensions that are not important for classification. For a given training sample, using larger models results in an increase of the variance of parameter estimates, which often degrades the ability of the classifier to perform well with new data not used during the learning process [41, 50].

Because of this, variable selection or dimension reduction are frequently added to the feature extraction to retain a smaller number of predictors and lower the size of the statistical models [50]. In common variable selection procedures, some coordinates of $\mathbf{X}$ are discarded and the remaining ones are retained without further processing [8]. On the other hand, dimension reduction typically involves some transformation of the features $\mathbf{X}$ followed by a selection process on the new coordinates to retain just a few of them [49].

A frequent choice with HMM-based classifiers is to use *linear* dimension reduction. In this type of reductions, a matrix $\boldsymbol{\rho} \in \mathbb{R}^{p \times d}, d \leq p$, is used to project the original features $\mathbf{X}$ onto a lower-dimensional subspace with coordinates $\boldsymbol{\rho}^T\mathbf{X} \in \mathbb{R}^d$. These d linear combinations of $\mathbf{X}$ should not lose any information carried by $\mathbf{X}$ that is relevant for classification. If successful, we could estimate models for $\boldsymbol{\rho}^T\mathbf{X}|Y$, instead of full-sized models for $\mathbf{X}|Y$.

The best known of linear reduction methods is principal component analysis (PCA). It seeks to maximize the variance of the retained coordinates as a way to conserve the information available in the original $\mathbf{X}$ [53]. However, PCA does not account for any dependence about between Y and $\mathbf{X}$, and thus important discriminative information can be lost in the reduction process. For a classification task, supervised dimension reduction is a better option. Examples of the most widely used methods in HMM-based classifiers are the subspace projection methods proposed in [56, 57, 83]. They pursue likelihood-based approaches to linear discriminant analysis (LDA) and heteroscedastic linear discriminant analysis (HLDA) for Gaussian data. As these methods are stated in a MLE framework, they can be consistently embedded into the training process of HMM. Nevertheless, both LDA and HLDA have been derived from heuristics, without taking care of retention of information.

Sufficient dimension reduction (SDR) is a relatively new approach that deals explicitly with loss of information for a particular objective [18, 60]. In a classification setting, $\boldsymbol{\rho}^T\mathbf{X}$ is said to be a linear sufficient reduction for $Y|\mathbf{X}$ if given $\boldsymbol{\rho}^T\mathbf{X}$ the class assignment is conditionally independent of the remaining information in $\mathbf{X}$ [23, 94]. However, SDR developments have been more tailored to regression problems, where the essential task is to estimate the smallest subspace of $\mathbf{X}$ that does not lose any information about Y. The sufficient reduction subspace in regression is usually larger than the sufficient discriminative subspace, but connections between them can be stated under some assumptions [23]. This allows us to use methods developed for regression in a classification framework.

The general SDR methodology does not require model assumptions for $\mathbf{X}$ or $\mathbf{X}|Y$ [16, 23, 60], but when a model for $\mathbf{X}|Y$ is assumed, maximum likelihood estimation can be used to estimate the fewest linear combinations of the features that retain all the information about Y. Existing model-based theory concerns conditional normal models only. It was introduced in [18] and further developed in [20, 21]. In particular, a new method called Likelihood Acquired Directions (LAD) was presented in [20] to deal with Gaussian data with unconstrained covariance. Nevertheless, these methods have not been explored neither for sequential data nor for complex classification tasks. In addition, understanding existing reduction methods for HMM-based classifiers under this framework is also pendant.

1.4 Contributions of the Thesis

This thesis deals with discriminative information when using HMMs for pattern recognition in sequential data. We focus on two different aspects:

- **Discriminative training of wavelet-domain HMM**.

  A new method for discriminative training of HMM-HMT models is introduced, aiming at improving the performance of sequential pattern recognizers in the wavelet domain. The proposed method relies in the MCE approach and provides reestimation formulas for fully non-tied models. An adapted version of Viterbi's decoding algorithm suited to HMM-HMT models is used to define the discriminant functions. Valued at each training sample, these functions are further combined in a single misclassification function whose sign determines the decision of the classifier. Direct application of standard procedures used with Gaussian mixture-HMMs is shown not to be effective for the HMM-HMT model, requiring a modification of the way rival candidates are weighted during the classification process. To deal with this, we propose a new approximation to the misclassification loss that penalizes differences in the order of magnitude of model likelihoods rather than in their values. As a result of this approximation, the updating process is driven not only by confusability of the training samples as is the usual approach, but also by the correctness of their classification. Phoneme recognition experiments with highly confusable phonemes from the TIMIT speech corpus [99] show that the proposed method consistently outperforms its MLE-based counterparts. Results from this contribution were published in [88, 89].

- **Sufficient dimension reduction of HMM**.
  Standard procedures for dimension reduction in HMM-based pattern recognizers are re-examined under the sufficiency approach. It is shown that both LDA and HLDA are capable of retaining all the class information of

the original features, but under quite strongs constraints on the covariance structure of the data that hardly ever hold in practice for a small dimension of the mantained subspace. As a consequence, to minimize the information loss HLDA usually needs to project the data to a subspace that is not the smallest one that could be obtained, thus losing efficiency. Most important, it is argued that LAD provides a better way to deal with heteroscedastic data, and that it outperforms HLDA when data is not constrained to the special covariance structure required by this method. A very special case arises if a reduction actually has a structured covariance as assumed in HLDA. The subspace estimated with HLDA may not be minimal even in this case, and the LAD estimator, albeit providing the smallest reduction yet, losses efficiency because it does not account for the special structure. We address this point and present a new estimator that both satisfies the same covariance structure as HLDA and gives a minimal sufficient reduction. On the other hand, the discussed theory allows us to derive methods to infer about the dimension of the smallest subspace that retains all the information to discriminate between the classes. This is useful in practice to serve as alternative to k-fold cross-validation or trial-and-error approaches. Developments are carried out for conditional normal models and its extension to HMM is shown. Results from this contribution have been reported in [86, 87], along with an open-access software toolkit for SDR methods published in [22].

1.5 Outline

We start by reviewing the basic theory and algorithms for HMM in Chapter 2. Both HMM with normal observation distributions and wavelet-domain HMM which use HMT as observation models are discussed. Contributions of the thesis are developed in Chapter 3 and Chapter 4. Concluding discussions are given in Chapter 5, along with further research derived from this work.

CHAPTER 2

Basics of hidden Markov models

2.1 Introduction

Hidden Markov models (HMM) are statistical models that haved proved useful to describe sequential data. They comprise a bivariate random process in which one of the variables forms a Markov chain. The state of the Markov chain remains hidden to the observer, but governs the outcome of the observed random variable in a probabilistic manner. The success of HMM lies in that they provide parsimonious parametric models for sequential data and in that there exist very efficient algorithms for estimating their parameters.

The basic theory on HMM was published by Baum and his colleagues [4]. Later, the proposed learning algorithms under the maximum likelihood framework turned out to be a special case of the expectation (EM) maximization algorithm for incomplete data [29]. In the applications area, they have shown to be remarkably useful for modeling speech, being at the core of automatic speech recognition, speech synthesis, spoken language understanding and machine translation [28, 48, 51, 80]. They have proved useful also in modeling and classification of proteins and genomic sequences [3], biomedical signals such as electrochardiograms [76], network traffic [25, 66] and econometric time-series [68]. In this chapter we review the basics of HMM, emphasizing the topics that will be needed in later developments. More comprehensive treatments can be found in [11, 37, 39, 51, 68].

2.2 Definition of HMM

Let $\{q_k\}$ be a sequence of random variables, with $k = 1, 2, \ldots, T$ and q_k taking values in the finite set $\{1, 2, \ldots, N_q\}$. Denote by $\mathbf{q}_1^t$ the subsequence $\{q_1, q_2, \ldots, q_t\}$. The sequence $\{q_k\}$ is said to form a *Markov chain* provided

$$\mathrm{p}(q_t|\mathbf{q}_1^{t-1}) = \mathrm{p}(q_t|q_{t-1}). \tag{2.1}$$

From the product rule of probability, the joint distribution of the overall sequence can be factorized as

$$\begin{aligned}
\mathrm{p}(\mathbf{q}_1^T) &= \mathrm{p}(q_T|\mathbf{q}_1^{T-1})\mathrm{p}(\mathbf{q}_1^{T-1}) \\
&= \mathrm{p}(q_T|\mathbf{q}_1^{T-1})\mathrm{p}(q_{T-1}|\mathbf{q}_1^{T-2})\mathrm{p}(\mathbf{q}_1^{T-2}) \\
&= \mathrm{p}(q_1)\prod_{t=1}^{T}\mathrm{p}(q_t|\mathbf{q}_1^{t-1}).
\end{aligned}$$

Thus, for a Markov chain we have

$$\mathrm{p}(\mathbf{q}_1^T) = \mathrm{p}(q_1)\prod_{t=1}^{T}\mathrm{p}(q_t|q_{t-1}). \tag{2.2}$$

If $\mathrm{p}(q_t = i|q_{t-1} = j)$ does not depend on the index t, the Markov chain is said to be *homogeneous* and it is completely specified by the set of parameters $\{\pi_i, a_{ij}\}$, with $\pi_i = \mathrm{p}(q_1 = i)$ and $a_{ij} = \mathrm{p}(q_t = i|q_{t-1} = j)$ for $i, j = 1, 2, \ldots, N_Q$. These parameters are constrained by

$$\begin{aligned}
\sum_{i=1}^{N_Q}\pi_i &= 1, \\
\sum_{i=1}^{N_Q}a_{ij} &= 1, \text{ for all } j.
\end{aligned} \tag{2.3}$$

Some state-transitions may not be allowed, so that $a_{ij} = 0$ for them. The set of allowed transitions, along with their corresponding probabilities, are often shown in a finite-state representation as the one shown in Figure 2.1. In this figure,

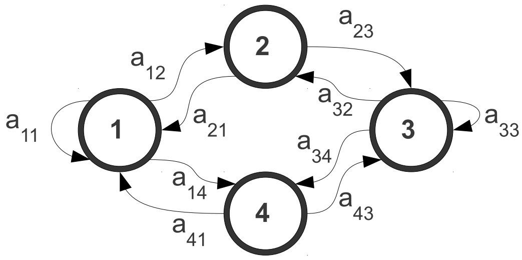


Figure 2.1: *Finite-state representation of a Markov chain. State-transtions* 1 → 3, 3 → 1, 2 → 4, 4 → 2, 2 → 2, *and* 4 → 4 *are not allowed for this example.*

for instance, the chain cannot jump neither between states 2 and 4, nor between states 1 and 3, nor stay in states 2 or 4 in consecutive instants.

Assume now that $\{q_k\}$ is not observable, but what is available to the observer is another sequence of random variables $\{X_k\}$ whose distribution is governed by the state of the Markov chain. In particular, assume that

$$\mathrm{p}(X_t|\mathbf{X}_1^{t-1}, \mathbf{q}_1^t) = \mathrm{p}(X_t|q_t), \tag{2.4}$$

with $\mathbf{X}_1^{t-1} = \{X_1, X_2, \ldots, X_{t-1}\}$. In this way, the distribution of X_t is determined only by q_t and it is conditionally independent of the remaining variables. For instance, X_t may be a normally distributed random variable whose mean and variance are determined by q_t.

When (2.1) and (2.4) hold, the random process $\{q_k, X_k\}$ is said to be a *hidden Markov chain.* In the engineering literature it is most commonly known as a *hidden Markov model* (HMM). The statistical dependence structure can be represented in a graphical model as the one shown in Figure 2.2. The graph summarizes that the observed variable X_t depends only on the hidden variable q_t and this depends only on the preceding q_{t-1}.

Assume t indexes time instants. At any t, the Markov chain takes a state $q_t = i$ out of the N_Q possible states and the observed output X_t is drawn from a probability density function $\mathrm{p}(X_t|q_t = i)$. At time $t+1$, the state of the chain can be the same as q_t, or it may have been evolved to other state $q_{t+1} = j$ according to a probability $\mathrm{p}(q_{t+1} = j|q_t = i)$. Given the state of the chain at this new instant, the output of the model is drawn now from $\mathrm{p}(X_{t+1}|q_{t+1} = j)$. The outputs $\{X_t\}$ are the unique observable quantities of the process, so the states $\{q_t\}$ of the underlying Markov chain always remain *hidden* to the observer.

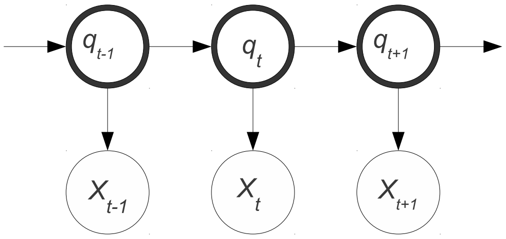


Figure 2.2: *Graphical-model representation of a HMM. The graph shows the statistical dependencies between the variables of the model.*

The observed variables X_t can be scalars or vectors, but the conditional distributions of $X_t|q_t$ are usually assumed to belong all to the same parametric family. We say that the HMM is homogeneous if the underlying Markov chain $\{q_t\}$ is homogeneous and the conditional distribution $X_t|q_t$ does not depend on the index t. In this case, the HMM is completely specified by the structure $\vartheta = \{\mathcal{Q}, \pi_i, a_{ij}, b_i(\cdot)\}$, where $\mathcal{Q} = \{1, 2, \ldots, N_Q\}$ is the set of allowed states for the latent variables q_t, $\pi_i = \mathrm{p}(q_1 = i|\vartheta)$ and $a_{ij} = \mathrm{p}(q_t = i|q_{t-1} = j, \vartheta)$ are the parameters of the underlying Markov chain $\{q_t\}$, and $b_i(\cdot)$ stands for the parametric model for $\mathrm{p}(X|q_t = i, \vartheta)$. Thus, given $\mathcal{Q}$, if N_b parameters are needed to characterize each observation model $b_i(X)$, in general we have $(1 + N_b + N_Q)N_Q$ parameters in the model that must satisfy the constraints (2.3). It is important to note that the observed sequence $\{X_k\}$ is not a Markov chain. In fact, one advantage of HMM relies in that they can model longer-range statistical dependencies between the observed variables through simple first-order dependencies between the latent variables $\{q_k\}$.

2.3 Model likelihood and computations

Let $\mathbf{X} = \mathbf{X}_1^T$ be a single sequence of observed features. Assume we model this sequence with an homogeneous HMM defined by the set of parameters ϑ and let $\mathbf{q} = \mathbf{q}_1^T$ be the sequence of states of the Markov chain at $t = 1, 2, \ldots, T$. As we cannot observe the sequence $\mathbf{q}$ that originated the observations, the likelihood $\mathcal{L}_\vartheta(\mathbf{X}) = \mathrm{p}(\mathbf{X}|\vartheta)$ accounts for all the possible paths $\mathbf{q}$ that could have generated the observed $\mathbf{X}$. Each path $\mathbf{q}$ has a joint probability $\mathrm{p}(\mathbf{X}, \mathbf{q})$. From assumptions (2.1) and (2.4), the likelihood then reads

$$\begin{aligned}\mathcal{L}_{\vartheta}(\mathbf{X}) &= \mathrm{p}(\mathbf{X}|\vartheta) \\ &= \sum_{\forall\mathbf{q}} \mathrm{p}(\mathbf{X}|\mathbf{q},\vartheta)\mathrm{p}(\mathbf{q}|\vartheta) \\ &= \sum_{\forall\mathbf{q}} \prod_{t=1}^{T} \mathrm{p}(X_t|q_t,\vartheta) \prod_{t=2}^{T} \mathrm{p}(q_t|q_{t-1},\vartheta)\mathrm{p}(q_1|\vartheta).\end{aligned}$$

Rearranging, we have

$$\mathcal{L}_{\vartheta}(\mathbf{X}) = \sum_{\forall\mathbf{q}} \mathrm{p}(q_1|\vartheta)\mathrm{p}(X_1|q_1,\vartheta) \prod_{t=2}^{T} \mathrm{p}(X_t|q_t,\vartheta)\mathrm{p}(q_t|q_{t-1},\vartheta),$$

where the summation is then over all possible sequences of states $\mathbf{q}$ that may have generated the observations. Using the notation introduced in Section 2.2 we get

$$\mathcal{L}_{\vartheta}(\mathbf{X}) = \sum_{\forall\mathbf{q}} b_{q_1}(X_1)\pi_{q_1} \prod_{t=2}^{T} b_{q_t}(X_t)a_{q_{t-1}q_t}. \tag{2.5}$$

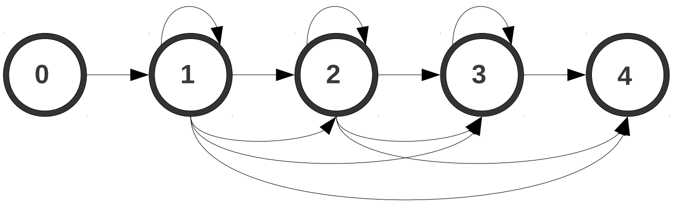
In many applications with sequential data, a particular type of HMM known as *left-to-right* HMM is used [28, 51, 80]. In this type of HMM, $a_{ij} = 0$ for $j > i$ and the initial state is fixed say at $q_1 = 1$ so that we can write $\pi_1 = a_{01} = 1$ and $\pi_j = 0$, $\forall j > 1$. Figure 2.3 shows a finite-state representation of this model and a corresponding trellis to show the possible paths that generated the observations. For this common structure we have

$$\mathcal{L}_{\vartheta}(\mathbf{X}) = \sum_{\forall\mathbf{q}} \prod_{t=1}^{T} b_{q_t}(X_t)a_{q_{t-1}q_t}. \tag{2.6}$$

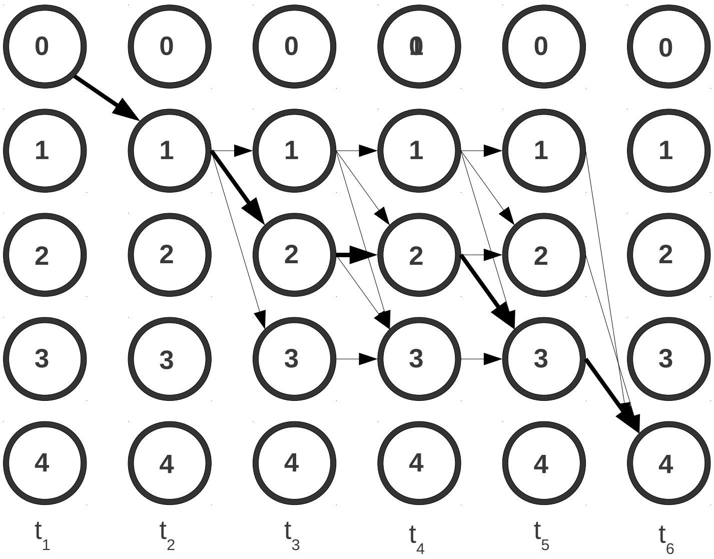
A key issue for the success of HMM is that there exist very efficient algorithms for computing the likelihood, for inference about the sequence of state that most likely generated the observations, and also for estimation of the parameters of the model [11, 80].

2.3.1 Parameter estimation

Likelihood computation assumes we know the parameters of model ϑ. In practice, we have to estimate them from the data. The usual framework to do



(a) Finite-state representation of a left-to-right HMM with five states.



(b) Trellis for a sequence of six observations modeled with the HMM in (a).

Figure 2.3: *a) Finite-state representation of a left-to-right HMM with five states. Note that states* 0 *and* 4 *are mandatory intial and final states, respectively. b) Trellis graph for a sequence of six observations modeled with a left-to-right HMM with five states. The arrows indicate the possible sequences of states taken by the underlying Markov chain to generate the obseved sequence* $\mathbf{X}_1^6 = \{X_1, X_2, \ldots, X_6\}$*. One of these paths is highlighted. Note that the chain can reach the final state* $q = 4$ *only at the final observation* X_6*.*

that is maximum likelihood estimation [80]. Estimating the likelihood directly is infeasible due to the large number of allowed sequences of states we would have to consider and the fact that each path includes the product of many probability factors that would lead to numerical underflow in computations. In addition, taking the logarithm of the likelihood function would not help, even when the conditional densities are taken from an exponential family, as it does not allow for any useful factorization.

A very efficient alternative is to use the **EM algorithm** for incomplete data [4, 29]. Assume for simplicity that we have a single observed sequence $\mathbf{X}$ to learn the parameters of ϑ. The sequence $\mathbf{X}$ is considered as *incomplete data*, being $(\mathbf{X}, \mathbf{q})$ the *complete* data [29]. The algorithm works iteratively. As $\mathbf{q}$ is not observed, it first estimates $\mathrm{p}(\mathbf{q}|\mathbf{X}, \vartheta^{old})$ in the E step, using the observed features and a current estimate of the model parameters ϑ^{old}. Given this estimation, in the M step the model parameters are updated by maximizing the expectation

$$\begin{aligned} \mathcal{Q}(\vartheta, \vartheta^{old}) &= \mathrm{E}_{\mathbf{q}|\mathbf{X},\vartheta^{old}} \{\log \mathrm{p}(\mathbf{q}, \mathbf{X}|\vartheta)\} \\ &= \sum_{\mathbf{q}} \mathrm{p}(\mathbf{q}|\mathbf{X}, \vartheta^{old}) \log \mathrm{p}(\mathbf{q}, \mathbf{X}|\vartheta). \end{aligned} \tag{2.7}$$

Maximizing this expectation amounts to maximizing (2.5), but computations are much more efficient since the joint likelihood $\log \mathrm{p}(\mathbf{q}, \mathbf{X}|\vartheta)$ factorizes conveniently.

To describe the computations in some detail, let us start by rewritting the expectation $\mathcal{Q}(\vartheta, \vartheta^{old})$ as

$$\begin{aligned} \mathcal{Q}(\vartheta, \vartheta^{old}) &= \sum_{\mathbf{q}} \sum_{t} \mathrm{p}(\mathbf{q}|\mathbf{X}, \vartheta^{old}) \log a_{q_{t-1}q_t} + \\ &\quad + \sum_{\mathbf{q}} \sum_{t} \mathrm{p}(\mathbf{q}|\mathbf{X}, \vartheta^{old}) \log b_{q_t}(X_t) \\ &= \sum_{i=1}^{N_Q} \sum_{j=1}^{N_Q} \sum_{t=1}^{T} \gamma_t(i,j) \log a_{ij} + \sum_{j=1}^{N_Q} \sum_{t=1}^{T} \gamma_t(j) \log b_j(X_t), \end{aligned}$$

where we have defined

$$\begin{aligned} \gamma_t(i,j) &\triangleq \mathrm{p}(q_{t-1} = i, q_t = j|\mathbf{X}, \vartheta^{old}) \\ &= \frac{\mathrm{p}(q_t = i, q_{t-1} = j, \mathbf{X}|\vartheta^{old})}{\mathrm{p}(\mathbf{X}|\vartheta^{old})} \end{aligned} \tag{2.8}$$

$$\begin{aligned} \gamma_t(j) &\triangleq \mathrm{p}(q_t = j|\mathbf{X}, \vartheta^{old}) \\ &= \frac{\mathrm{p}(q_t = j, \mathbf{X}|\vartheta^{old})}{\mathrm{p}(\mathbf{X}|\vartheta^{old})}. \end{aligned} \tag{2.9}$$

Then, in the E step we compute the quantities $\gamma_t(i,j)$ and $\gamma_t(j)$ using a current estimate ϑ^{old} of the model parameters, and use these results in the M step to update the model parameters by maximizing $\mathcal{Q}(\vartheta, \vartheta^{old})$.

The E step. The efficient implementation of this step of the algorithm requires the definition of a pair of auxiliary variables that can be computed recursively. Define the *forward* variable

$$\alpha_t(i) \triangleq \mathrm{p}\left(\mathbf{X}_1^t, q_t = i \,|\vartheta\right). \tag{2.10}$$

Starting with $\alpha_1(i) = \pi_i b_i(X_1)$, it is shown that it can be computed with the recurssion [6, 80]

$$\alpha_t(i) = b_i(X_t) \sum_{j=1}^{N_Q} \alpha_{t-1}(j) a_{ji}. \tag{2.11}$$

Similarly, we can define a *backward* variable

$$\beta_t(i) \triangleq \mathrm{p}\left(\mathbf{X}_{t+1}^T \,|q_t = i, \vartheta\right). \tag{2.12}$$

Starting from $\beta_T(i) = 1/N_Q$, it is also shown that it can computed recursively as [6, 80]

$$\mathrm{p}(\mathbf{X}_{t+1}^T|q_t = i, \vartheta) = \sum_{j=1}^{N_Q} b_j(X_{t+1})\beta_{t+1}(j)a_{ij}. \tag{2.13}$$

From definitions (2.10) and (2.12), we see that $\gamma_t(i,j)$ and $\gamma_t(j)$ can be computed efficiently as

$$\begin{aligned} \gamma_t(i,j) &= \frac{\alpha_{t-1}(i)a_{ij}b_j(X_t)\beta_t(j)}{\sum_{j=1}^{N_Q}\alpha_T(j)}, & (2.14)\\ \gamma_t(j) &= \frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N_Q}\alpha_T(j)}. & (2.15) \end{aligned}$$

Here we have used the recurssion for the forward variable to compute the likelihood of the observed sequence $\mathbf{X}$ under model ϑ^{old}

$$\mathcal{L}_{\vartheta^{old}}(\mathbf{X}) = \mathrm{p}(\mathbf{X}|\vartheta^{old}) = \sum_{\forall i \in Q} \mathrm{p}(\mathbf{X}, q_T = i|\vartheta^{old}) = \sum_{\forall i \in Q} \alpha_T(i). \tag{2.16}$$

The M step. Once the E step has been completed, the model parameters are updated by maximizing $\mathcal{Q}(\vartheta, \vartheta^{old})$. Note first that

$$\begin{aligned}\mathcal{Q}(\vartheta, \vartheta^{old}) &= \sum_{i=1}^{N_Q}\sum_{j=1}^{N_Q}\sum_{t=1}^{T}\gamma_t(i,j)\log a_{ij} + \sum_{j=1}^{N_Q}\sum_{t=1}^{T}\gamma_t(j)\log b_j(X_t) \\ &= \mathcal{Q}_a(\vartheta, \vartheta^{old}) + \mathcal{Q}_b(\vartheta, \vartheta^{old}).\end{aligned}$$

As $\mathcal{Q}_b(\vartheta, \vartheta^{old})$ does not depend on the state-transition probabilities a_{ij} after the quantities $\gamma_t(i,j)$ have been obtained in the E step, the estimation of parameters $\{a_{ij}\}$ requires the maximization of just $\mathcal{Q}_a(\vartheta, \vartheta^{old})$. As a consequence, the estimation of the state-transition probabilities has the same form regardless the choice of parametric observation models. Maximizing $\mathcal{Q}_a(\vartheta, \vartheta^{old})$ with the constraints (2.3) leads to the set of re-estimation formulas

$$a_{ij} = \frac{\sum\limits_{t=1}^{T}\gamma_t(i,j)}{\sum\limits_{t=1}^{T}\sum\limits_{j=1}^{N_Q}\gamma_t(i,j)}, \text{ for } i,j = 1,2,\ldots,N_Q. \tag{2.17}$$

Likewise, updating the parameters of the observation models $\{b_i(X)\}$ requires the maximization of $\mathcal{Q}_b(\vartheta, \vartheta^{old})$ only, but to derive the specific re-estimation formulas we have to assume a parametric model for $b_i(X)$. In the next paragraph we describe this step when the observation models are normal densities.

Gaussian HMM. In many HMM applications, the observations are random vectors of features $X_t = \mathbf{x}_t \in \mathbb{R}^p$ and multivariate normal densities or mixtures of normal densities are used as observation models. We will refer to this models as *normal* hidden Markov models or simply as **GHMM**. For simplicity, assume $b_j(\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_j, \boldsymbol{\Delta}_j)$. In this case, we have

$$\begin{aligned}\mathcal{Q}_b(\vartheta, \vartheta^{old}) &= \sum_{j=1}^{N_Q}\sum_{t=1}^{T}\gamma_t(j)\log b_j(\mathbf{x}_t) \\ &= -\frac{1}{2}\sum_{j=1}^{N_Q}\sum_{t=1}^{T}\gamma_t(j)(\mathbf{x}_t-\boldsymbol{\mu}_j)^T\boldsymbol{\Delta}_j^{-1}(\mathbf{x}_t-\boldsymbol{\mu}_j) + B,\end{aligned}$$

where

$$B = -\frac{1}{2}\sum_{j=1}^{N_Q}\sum_{t=1}^{T}\gamma_t(j)[p\log(2\pi) + \log|\boldsymbol{\Delta}_j|].$$

Maximizing with respect to $\boldsymbol{\mu}_j$ and $\boldsymbol{\Delta}_j$ we get [6, 80]

$$\boldsymbol{\mu}_j = \frac{\sum\limits_{t=1}^{T}\gamma_t(j)\mathbf{x}_t}{\sum\limits_{t=1}^{T}\gamma_t(j)} \tag{2.18}$$

$$\boldsymbol{\Delta}_j = \frac{\sum\limits_{t=1}^{T}\gamma_t(j)(\mathbf{x}_t - \boldsymbol{\mu}_j)(\mathbf{x}_t - \boldsymbol{\mu}_j)^T}{\sum\limits_{t=1}^{T}\gamma_t(j)}. \tag{2.19}$$

Remember that in these derivations we have considered the likelihood for a single long sequence of observations (see 2.5). In machine learning applications, we typically have a set of observations $\{\mathbf{X}_p\}$, with $p = 1, 2, \ldots, P$, to learn the parameters for each model. In this case, the usual assumption is that each observed sequence is statistically independent of the others, so that the obtained formulas simply take the form

$$\boldsymbol{\mu}_j = \frac{\sum\limits_{p=1}^{P}\sum\limits_{t=1}^{T_p}\gamma_t^p(j)\mathbf{x}_t^p}{\sum\limits_{p=1}^{P}\sum\limits_{t=1}^{T_p}\gamma_t^p(j)} \tag{2.20}$$

$$\boldsymbol{\Delta}_j = \frac{\sum\limits_{p=1}^{P}\sum\limits_{t=1}^{T_p}\gamma_t^p(j)(\mathbf{x}_t^p - \boldsymbol{\mu}_j)(\mathbf{x}_t^p - \boldsymbol{\mu}_j)^T}{\sum\limits_{p=1}^{P}\sum\limits_{t=1}^{T_p}\gamma_t^p(j)}. \tag{2.21}$$

A deeper view to the EM algorithm. In previous paragraphs we described how the EM algorithm works iteratively on an auxiliary function $\mathcal{Q}(\vartheta, \vartheta^{old})$ in order to maximize the likelihood function $\mathcal{L}_\vartheta(\mathbf{X})$. A nice presentation of this relationship is given in [6]. To start with, let $g(\mathbf{q})$ be a distibution defined over the latent (hidden) variables and assume $g(\mathbf{q}) > 0$. For any choice of $g(\mathbf{q})$, we can rewrite the logarithm of the likelihood as [1]

$$\begin{aligned} \log \mathcal{L}_\vartheta(\mathbf{X}) &= \log \mathrm{p}(\mathbf{X}|\vartheta), \\ &= \sum_{\mathbf{q}} g(\mathbf{q}) \log \frac{\mathrm{p}(\mathbf{X},\mathbf{q}|\vartheta)}{g(\mathbf{q})} - \sum_{\mathbf{q}} g(\mathbf{q}) \log \frac{\mathrm{p}(\mathbf{q}|\mathbf{X},\vartheta)}{g(\mathbf{q})}, \\ &= \mathbb{L}(g,\vartheta) + \mathrm{KL}(\mathrm{p}(\mathbf{q}|\mathbf{X},\vartheta)||g), \end{aligned}$$

where

$$\begin{aligned} \mathbb{L}(g,\vartheta) &= \sum_{\mathbf{q}} g(\mathbf{q}) \log \frac{\mathrm{p}(\mathbf{X},\mathbf{q}|\vartheta)}{g(\mathbf{q})}, \\ \mathrm{KL}(\mathrm{p}(\mathbf{q}|\mathbf{X},\vartheta)||g) &= -\sum_{\mathbf{q}} g(\mathbf{q}) \log \frac{\mathrm{p}(\mathbf{q}|\mathbf{X},\vartheta)}{g(\mathbf{q})}. \end{aligned}$$

In the last expression, $\mathrm{KL}(\mathrm{p}(\mathbf{q}|\mathbf{X},\vartheta)||g)$ is the Kullback-Leibler divergence between $g(\mathbf{q})$ and the posterior distribution $\mathrm{p}(\mathbf{q}|\mathbf{X},\vartheta)$. This term is nonnegative. Thus, $\log \mathrm{p}(\mathbf{X}|\vartheta) \geq \mathbb{L}(g,\vartheta)$ and, $\mathbb{L}(g,\vartheta)$ is a lower bound for $\log \mathrm{p}(\mathbf{X}|\vartheta)$. With these ingredients, we can think of a general EM algorithm as a two-step iterative process where we seek to maximize the log-likelihood $\log \mathrm{p}(\mathbf{X}|\vartheta)$ *by maximizing the lower bound* $\mathbb{L}(g,\vartheta)$ [6]:

- In the E step of the EM algorithm, the bound is maximized over $g(\mathbf{q})$ while holding fixed the current estimate of the model parameters ϑ^{old}. When ϑ^{old} is fixed, the likelihood $\log \mathrm{p}(\mathbf{X}|\vartheta^{old})$ is fixed and the maximum of the bound occurs at $\mathrm{KL}(\mathrm{p}(\mathbf{q}|\mathbf{X},\vartheta)||g) = 0$, which gives $g(\mathbf{q}) = \mathrm{p}(\mathbf{q}|\mathbf{X},\vartheta^{old})$.

- In the M step, $g(\mathbf{q})$ is held fixed at $g(\mathbf{q}) = \mathrm{p}(\mathbf{q}|\mathbf{X},\vartheta^{old})$ and the lower bound is maximized with respect to ϑ to update the current estimate ϑ^{old}. This step will cause $\mathbb{L}(g,\vartheta)$ to increase, unless it is already at a maximum. With these new estimates, we expect $\mathrm{KL} > 0$ since the model parameters have changed from ϑ^{old} and thus $\log \mathrm{p}(\mathbf{X}|\vartheta) > \log \mathrm{p}(\mathbf{X}|\vartheta^{old})$.

Iterations are repeated until convergence. This general view of the EM algorithm has a broader scope than we need here. But what is interesting to note is that

[1] To see this, decompose $\sum_{\mathbf{q}} g(\mathbf{q}) \log \frac{\mathrm{p}(\mathbf{X},\mathbf{q}|\vartheta)}{g(\mathbf{q})}$ and note that $\sum_{\mathbf{q}} g(\mathbf{q}) \log \mathrm{p}(\mathbf{X}|\vartheta) = \log \mathrm{p}(\mathbf{X}|\vartheta)$.

since $g(\mathbf{q})$ is fixed at $\mathrm{p}(\mathbf{q}|\mathbf{X},\vartheta)$ in the M step, the lower bound reads

$$\begin{aligned}\mathbb{L}(g,\vartheta) &= \sum_{\mathbf{q}} \mathrm{p}(\mathbf{q}|\mathbf{X},\vartheta^{old})\log \mathrm{p}(\mathbf{X},\mathbf{q}|\vartheta) - \\ &\quad - \sum_{\mathbf{q}} \mathrm{p}(\mathbf{q}|\mathbf{X},\vartheta^{old})\log \mathrm{p}(\mathbf{q}|\mathbf{X},\vartheta^{old}) \\ &= \mathcal{Q}(\vartheta,\vartheta^{old}) + \text{const.}\end{aligned}$$

Thus, maximizing $\mathcal{Q}(\vartheta,\vartheta^{old})$ as we did in our presentation of the EM algorithm for hidden Markov models, is the same as maximizing $\mathbb{L}(g,\vartheta)$ at the M step and then we see that maximizing $\mathcal{Q}(\vartheta,\vartheta^{old})$ amounts to maximizing the log-likelihood $\log \mathrm{p}(\mathbf{X}|\vartheta)$.

2.3.2 Inference: Viterbi's algorithm

The forward-backward recursions reviewed in Section 2.3.1 provide an efficient way to compute the likelihood of an observed sequence given the model ϑ. Nevertheless, in many cases we are interested in infering about the sequence of states which is more likely to have generated the observed data. This amounts to find the sequence $\tilde{\mathbf{q}}$ that maximizes the joint likelihood $\mathrm{p}(\mathbf{X},\mathbf{q})$, so that [6, 48]

$$\tilde{\mathbf{q}} = \arg\max_{\mathbf{q}} \prod_{t=1}^{T} b_{q_t}(X_t) a_{q_{t-1}q_t} \tag{2.22}$$

The algorithm that efficiently optimizes this search is known as *Viterbi's algorithm*. We can think of it as a modification of the forward algorithm, in which instead of summing up probabilities from different paths coming to the same destination state (see 2.10), only the best path is picked and remembered.

To do this, define an auxiliary variable $\lambda_t(j)$ as

$$\lambda_t(j) \triangleq \max_{\forall \mathbf{q}^{t-1}} \left\{ \mathrm{p}\left(\mathbf{q}^{t-1}, q_t = j, \mathbf{X}^t \,|\vartheta\right)\right\}; \quad \forall j \in Q. \tag{2.23}$$

Similarly to the forward variable, starting with $\lambda_1(i) = \pi_i b_i(X_1)\ \forall i \in Q$, it can be computed with the recursion

$$\begin{aligned}\lambda_t(j) &= \max_{1\le i\le N_Q} \{\lambda_{t-1}(i)\mathrm{p}(q_t = j|q_{t-1} = i,\vartheta)\mathrm{p}(X_t|q_t = j, q_{t-1} = i,\vartheta)\} \\ &= \max_{1\le i\le N_Q} \{\lambda_{t-1}(i)\mathrm{p}(q_t = j|q_{t-1} = i,\vartheta)\mathrm{p}(X_t|q_t = j,\vartheta)\} \\ &= \max_{1\le i\le N_Q} \{\lambda_{t-1}(i)a_{ij}\} b_j(X_t).\end{aligned} \tag{2.24}$$

Parallel to this variable, define:

$$\xi_t(j) \triangleq \arg\max_{\forall i \in Q} \{\lambda_{t-1}(i) a_{ij}\}.$$

Thus, from

$$\tilde{q}_T = \arg\max_{\forall i \in Q} \{\lambda_T(i)\}$$

we obtain the best path $\tilde{\mathbf{q}}$ using the inverse recursion:

$$\tilde{q}_t = \xi_{t+1}(\tilde{q}_{t+1}); \quad t = T-1, T-2, \dots, 1$$

In many cases, the best score $\max_{j \in Q} \lambda_T(j) = \mathrm{p}(\mathbf{X}, \tilde{\mathbf{q}})$ is a good approximation to the (complete) likelihood $\mathrm{p}(\mathbf{X}|\vartheta)$, and it is then used for classification.

2.4 Hidden Markov models in the wavelet domain

Multiscale analysis using wavelets is a well-established tool for signal and image representation [27, 67]. The multiresolution property of the wavelet transform and its flexibility to deal with local features simultaneously in time/space and frequency provide a suitable scenario for many signal processing and pattern recognition tasks. Initial interest in these representations was largely driven by powerful non-linear methods which relied on simple scalar transformations of coefficients [31]. Many posterior developments kept in mind the idea of some decorrelation property of the wavelet transform or assumed very simple statistical models for the coefficients. Nevertheless, in practical applications signals and images usually show sparse representations and some structural dependence between coefficients which cannot be described with such models. Simply speaking, coefficients typically are not normally distributed and large ones tend to form clusters along scales and to propagate across scales [67]. Because of this, both coefficients magnitude and statistical dependencies between them carry relevant information about signals and their underlying distribution.

These features can be exploited for pattern recognition, but the joint distribution of the coefficients is needed. While complete knowledge of this probability is infeasible, we can replace it with a suitable model that accounts for the main properties of the representation while remaining simple enough and computationally tractable. If we succeed in doing this, we can use these models straightforwardly for statistical pattern recognition, without the need of specific feature extraction procedures that can loss important information.

2.4.1 The discrete wavelet transform

We measure a signal with the aim to extract some useful information from it. The measurement process is usually done in a way that is convenient technologically, but the information within the measured signal can be difficult to interprete. Thus, we look for a transformation of the signal so that the new *representation* allows us to easily extract the information.

Wavelet analysis has shown to provide useful representations of signals and images in many applications. There are several different transforms commonly grouped as wavelet transforms [67]. In all of them, each coefficient or *atom* of the decomposition provides a local weighted average of the signal at a certain scale and interval of time. Thus, we can think of these transforms as providing a mapping of a signal onto a time-scale plane. Different wavelet transforms differ in the partition they induce on that plane.

In this thesis we work with the DWT, which provides an orthogonal decomposition for vectors in $\mathbb{R}^N$. It can be computed very efficiently [67] and induces a *dyadic* partition of the time-scale plane that allows for representing the obtained coefficients naturally as a binary tree. This structure helps to make computations very efficient, which is an important factor in applications.

To briefly describe this transformation, assume $\mathbf{z} \in \mathbb{R}^N$, with $N = 2^J$, is the sampled measured signal[2]. The DWT of $\mathbf{z}$ is $\mathbf{w} = \mathcal{W}\mathbf{z}$, where $\mathcal{W}$ is an $N \times N$ matrix defining the transformation and satisfying $\mathcal{W}^T\mathcal{W} = \mathbf{I}_N$. Particular values for this matrix depends on the wavelet filters chosen for the analysis. The n-th coefficient of $\mathbf{w}$, w_n, is a local average over a particular scale and a particular set of times. From the orthogonality of the transform, w_n^2 measures the energy of the signal at that scale and interval of times. Then, $\mathbf{w}$ represents a multiresolution decomposition of $\mathbf{z}$ at scales $\tau_j = 2^{j-1}$, for $j = 1, 2, \ldots, J$. The analysis gives $N/(2\tau_j)$ coefficients at each scale and they can be arranged so that coefficients

[2] The condition that the length of $\mathbf{z}$ be a power of two is too restrictive and can be remove in practice, but we keep it here for ease of exposition.

belonging to the same scale of analysis are adjacent in $\mathbf{w}$. Furthermore, two adyacents rows of $\mathcal{W}$ that corresponds to the same scale j are circularly shifted versions of each other by an amount 2^j.

Computations are perfomed very efficiently using the pyramidal algorithm [67]. The obtained representations tend to be *sparse*, meaning that a few coefficients concentrate most of the energy of the signal. From an statistical point of view, it means that if we regard the coefficients as realizations of a random process, their marginal density is often very sharp near zero, which leads to a distribution with a kurtosis greater than for the normal density.

Another key property of the wavelet transform is *locality*. It accounts to the fact that each atom of the decomposition is concentrated simultaneously in time and in scale/frequency. As stated above, each coefficient carries the energy of the signal in a given region of the time-scale plane. The tilling of the plane induced by the DWT is shown in Figure 2.4, with each rectangle being related to a given coefficient in the representation. Note that the area of the rectangles is constant for all of them. If we colour the rectangles according to the squared magnitud of the associated coefficients, we obtain a graph known as *scalogram*. A main feature of the wavelet representations of real-world signals and images is that this graph often shows *clusters* of coeffcients for which their magnitud is large, as well as this trend in intensity tending to propagate across scales, something that is frequently referred to as the *persistance* property of the transform.

If we are to use some statistical model of the wavelet coefficients, we should account for the properties just discussed. We discuss next a parsimonious model that does this.

2.4.2 Hidden Markov trees

Crouse et al. [24] proposed a multiresolution Markov model to concisely account for properties of wavelet representations of signals and images. In their framework, the marginal probability of each coefficient is modeled as a Gaussian mixture driven by a hidden state variable. While the mixture accounts for sparseness, markovian relationships between hidden states allow for describing dependencies between coefficients. The resulting structure is then a hidden Markov model on the wavelet domain which exploits the natural tree structure of DWT,

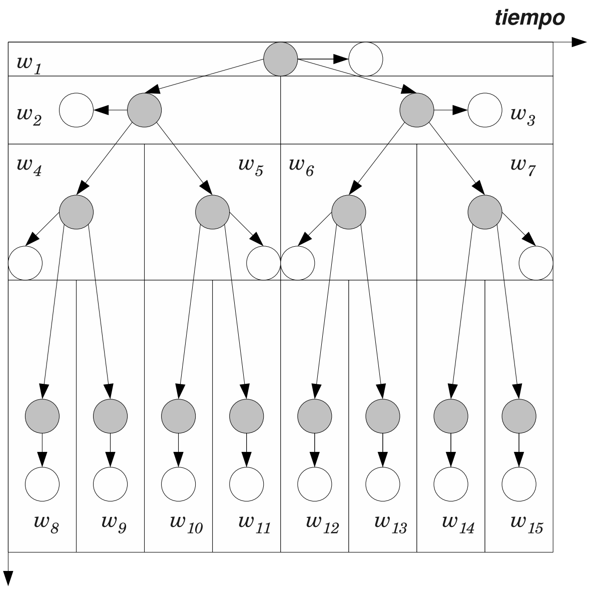


Figure 2.4: *Eschematics of the HMT model. A hidden latent variable (shadowed circles) is associated to each rectangle in the time-frequency plane and Markovian dependencies are set between them. The state of the latent variable determines the parameters of the normal distributions related to the observed coefficients linked to them. Thus, the observed coefficients (white circles) are assumed conditional independent of the other variables and their marginal distribution is a mixture of normal densities.*

and it is usually referred to as hidden Markov tree (HMT). Figure 2.4 shows a diagram of the model. Other multiresolution Markov models are reviewed in [91], with an emphasis in signal and image processing. Some of them do not use latent variables but set statistical dependencies between wavelet coefficients directly. Nevertheless, many of these models are targetted to specific applications and can be described only in those contexts. Throughout this thesis we will focus only in the HMT, which has been found useful in a broad range of applications concerning both signals and images.

Let $\mathbf{w} = [w_1, w_2, \dots, w_N]$, with $w_u \in \mathbb{R}$, be the observed features, which result from a DWT analysis of the signal with J scales and discarding w_0, the approximation coefficient at the coarsest scale. From the partition of the time-scale plane induced by the transformation, the random vector of coefficients $\mathbf{w}_t$ can also be indexed as a tree rooted in w_1. Associated with each wavelet coefficient, there is a latent (hidden) variable r_u. Thus, associated with the vector of coefficients $\mathbf{w}$ there is a vector of hidden states $\mathbf{r} = [r_1, r_2, \dots, r_N]$ that can also be indexed as a tree rooted in r_1. Each latent variable r_u takes values in the set $\{1, 2, \dots, K\}$. We will usually refer to $u = 1, 2, \dots, N$ as *nodes*. For $u = 2, \dots, N$, $\rho(u)$ will denote the parent node of u. In addition, if u is not a leaf of the tree structure, $\mathcal{C}_u = \{c_1(u), \dots, c_{N_u}(u)\}$ will denote the set of children nodes of u. Note that for a dyadic tree resulting from a DWT analysis, each non-terminal node has two children. These variables are said to define a HMT provided they fulfill the following assumptions [34]:

1. $\forall u \in \{1, 2, \dots, N\}$, the marginal distribution of w_u is a mixture

$$\mathrm{p}(w_u = w) = \sum_{k=1}^{K} \mathrm{p}(r_u = k) f_{u,k}(w),$$

   where $f_{u,k}(w_u) = \mathrm{p}(w_u | r_u = k)$.

2. Markov tree property for the latent variables

$$\mathrm{p}(r_u = m | \{r_v / v \neq u\}) = \mathrm{p}(r_u = m | r_{\rho(u)}).$$

3. The observed coefficients depend on the state of the latent variables, not on the rest of coefficients

$$\mathrm{p}(w_1, \dots, w_N | r_1, \dots, r_N) = \prod_{u=1}^{N} \mathrm{p}(w_u | r_1, \dots, r_N).$$

4. The observed coefficients depend only on the state of the latent variable associated to them in the corresponding node of the tree

$$\mathrm{p}(w_u | r_1, r_2, \dots, r_N) = \mathrm{p}(w_u | r_u), \ \forall u.$$

Note that the last two assumptions resemble the conditional independence property of usual HMM as discussed in Section 2.2. The dependence structure of the HMT is shown in Figure 2.5.

Similarly to a conventional HMM, the HMT is characterized for the set of parameters $\theta = (\{\kappa_m\}, \{\epsilon_{u,mn}\}, \{f_{u,m}\})$, where $\kappa_m = \mathrm{p}(r_1 = m | \theta)$, $\epsilon_{u,mn} = \mathrm{p}(r_u = m | r_{\rho(u)} = n, \theta)$, and $f_{u,m} = \mathrm{p}(w_u | r_u = m, \theta)$ as defined previously. Usually, $f_{u,m}$ is assumed normal. Despite the similarities with conventional HMM, there are some important differences between them and HMT that are important to note. First, there is not a temporal notion in the HMT. All wavelet coeffcients are observed

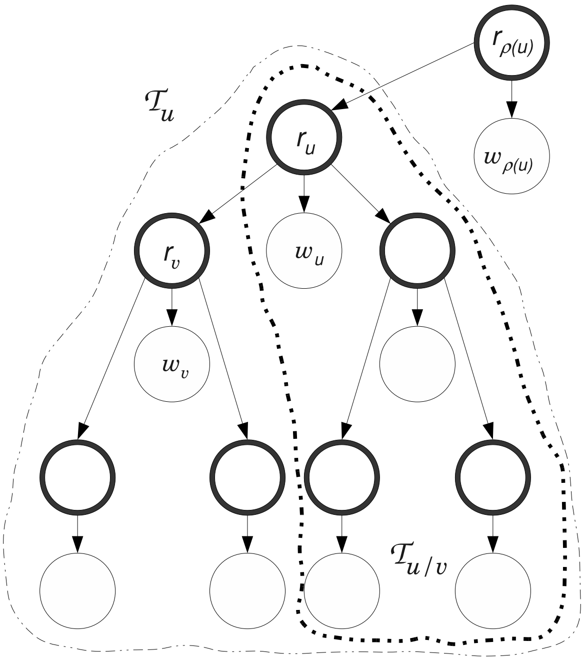


Figure 2.5: *Graphical-model representation of a HMT. Only a part of the tree is shown.*

simultaneously. Second, though the state-transition probabilities $\epsilon_{u,mn}$ are often assumed independent of the node u, this assumption is usually stronger than in the homogeneous conventional HMM and aims mainly at reducing the number of parameters in the model. This is an example of strong parameter tying that is found frequently in signal processing applications. Nevertheless, in machine learning we often have a set of training signals for parameter estimation and we can hope to learn larger models keeping the variance of parameter estimates acceptable.

Likelihood of the HMT. From the assumptions stated above, the likelihood $\mathcal{L}_\theta(\mathbf{w}) = \mathrm{p}(\mathbf{w}|\theta)$ for the HMT model reads [24]

$$\begin{aligned} \mathcal{L}_\theta(\mathbf{w}) &= \mathrm{p}(w_1, \ldots, w_N|\theta) \\ &= \sum_{\forall \mathbf{r}} \mathrm{p}(r_1, \ldots, r_N, w_1, \ldots, w_N|\theta) \\ &= \sum_{\forall \mathbf{r}} \mathrm{p}(r_1, \ldots, r_N|\theta)\mathrm{p}(w_1, \ldots, w_N|r_1, \ldots, r_N, \theta), \end{aligned} \tag{2.25}$$

where the summation is over all possible combinations of states $\mathbf{r}$ in the nodes of the tree. The first factor in each term of the summation represents the probability of each of those combinations of states. From the Markov property of the tree, we have

$$
\begin{aligned}
\mathrm{p}(r_1, \ldots, r_N|\theta) &= \mathrm{p}(r_1|\theta) \prod_{u=2}^{N} \mathrm{p}(r_u|r_{\rho(u)}, \theta) \\
&= \pi_{r_1} \prod_{u=2}^{N} \epsilon_{u,r_u r_{\rho(u)}} \qquad (2.26)
\end{aligned}
$$

The second factor in each term of the summation can be simplified using the conditional independence assumptions for the HMT, reading

$$
\begin{aligned}
\mathrm{p}(w_1, \ldots, w_N|r_1, \ldots, r_N, \theta) &= \prod_{u=1}^{N} \mathrm{p}(w_u|r_1, \ldots, r_N, \theta) \\
&= \prod_{u=1}^{N} \mathrm{p}(w_u|r_u, \theta) \\
&= \prod_{u=1}^{N} f_{u,r_u}(w_u). \qquad (2.27)
\end{aligned}
$$

Replacing back in the likelihood and letting $\epsilon_{1,r_1 r_{\rho(1)}} = \pi_{r_1}$, we get

$$
\begin{aligned}
\mathcal{L}_\theta(\mathbf{w}) &= \sum_{\forall \mathbf{r}} \pi_{r_1} \prod_{u=2}^{N} \epsilon_{u,r_u r_{\rho(u)}} \prod_{u=1}^{N} f_{u,r_u}(w_u) \\
&= \sum_{\forall \mathbf{r}} \prod_{u=1}^{N} \epsilon_{u,r_u r_{\rho(u)}} f_{u,r_u}(w_u). \qquad (2.28)
\end{aligned}
$$

We see that this expression for the likelihood of the HMT resembles that for the standard HMM. Nevertheless, we must keep in mind that transition probabilities in the time-domain HMM have very different meaning than time-scale transitions in the HMT.

As with conventional HMM, there are three basic problems related to the HMT: efficient likelihood computation; parameter estimation; and inference of the best combination of states for the latent variables in the tree.

Parameter estimation. Parameters in the HMT model θ are estimated using an adapted EM algorithm [24, 34]. To start with, note that maximizing the likelihood given a learning set $\{\mathbf{w}^\ell\}$ of independent random vectors $\mathbf{w}^\ell$ is identical to iteratively maximizing the auxiliary function

$$\begin{aligned}\mathcal{Q}(\theta,\theta^{old})(\{\mathbf{w}^\ell\}) &= \mathrm{E}_{\mathbf{w}}\left\{\mathrm{E}_{\mathrm{p}(\mathbf{r}|\mathbf{w}^\ell,\theta^{old})}\left\{\log \mathrm{p}(\mathbf{w}^\ell,\mathbf{r}|\theta)\right\}\right\}\\ &= \sum_\ell\sum_u\sum_m\sum_n \xi_u^\ell(m,n)\log\epsilon_{u,mn} + \\ &\quad + \sum_\ell\sum_u\sum_m \gamma_u^\ell(m)\log f_{u,m}(w_u^\ell),\end{aligned}$$

where we have used the definitions

$$\begin{aligned}\xi_u^\ell(m,n) &\triangleq \mathrm{p}(r_u=m, r_{\rho(u)}=n|\mathbf{w}^\ell,\theta^{old}) \qquad (2.29)\\ &= \frac{\mathrm{p}(\mathbf{w}^\ell, r_u=m, r_{\rho(u)}=n|\theta)}{\mathrm{p}(\mathbf{w}^\ell|\theta)},\\ \gamma_u^\ell(m) &\triangleq \mathrm{p}(r_u=m|\mathbf{w}^\ell,\theta^{old}) \qquad (2.30)\\ &= \frac{\mathrm{p}(\mathbf{w}^\ell, r_u=m|\theta)}{\mathrm{p}(\mathbf{w}^\ell|\theta)}.\end{aligned}$$

The E step involves computing $\xi_u^\ell(m,n)$ and $\gamma_u^\ell(m)$. This can be done efficiently using upward and downward recursions through the tree that are defined similarly to the forward and backward variables described for conventional HMM. The algorithm was first proposed in [24] and improved in [34]. We describe it in a way it is easy to compare it with the conventional forward-backward algorithm. Further details are given in [34].

Let $\mathcal{T}_u$ be the subtree of observed wavelet coefficients rooted in node u, so that $\mathcal{T}_1$ is the complete observed tree, and let $\mathcal{T}_{u\setminus v}$ be the subtree rooted in u so that the coefficients in $\mathcal{T}_v$ are also in $\mathcal{T}_u$ but not in $\mathcal{T}_{u\setminus v}$ (see Figure 2.5). Define

$$\begin{aligned}\alpha_u(n) &\triangleq \mathrm{p}\left(\mathcal{T}_{1\setminus u}, r_u=n\,|\theta\right), \qquad (2.31)\\ \beta_u(n) &\triangleq \mathrm{p}\left(\mathcal{T}_u\,|r_u=n,\theta\right), \qquad (2.32)\\ \beta_{\rho(u),u}(n) &\triangleq \mathrm{p}\left(\mathcal{T}_u\,\middle|r_{\rho(u)}=n,\theta\right). \qquad (2.33)\end{aligned}$$

Variables $\beta_u(n)$ and $\beta_{\rho(u),u}(n)$ are computed recursively going upward through the tree from the leaves to the root node, while $\alpha_u(n)$ is computed recursively going downwards throughout the tree. The recursions can be obtained as [24]

$$\begin{aligned}
\beta_u(n) &= \mathrm{p}\left(\mathcal{T}_u \left| r_u = n, \theta\right.\right) \\
&= \left\{ \prod_{v \in C(u)} \mathrm{p}\left(\mathcal{T}_v | r_u = n, \theta\right) \right\} \mathrm{p}\left(w_u | r_u = n, \theta\right) \\
&= \left\{ \prod_{v \in C(u)} \beta_{u,v}(n) \right\} f_{u,n}(w_u), \qquad (2.34)
\end{aligned}$$

where we have

$$\begin{aligned}
\beta_{\rho(u),u}(n) &= \mathrm{p}\left(\mathcal{T}_u \left| r_{\rho(u)} = n, \theta\right.\right) \\
&= \sum_{m=1}^{M} \mathrm{p}\left(\mathcal{T}_u | r_u = m, \theta\right) \mathrm{p}\left(r_u = m | r_{\rho(u)} = n, \theta\right) \\
&= \sum_{m=1}^{M} \beta_u(m) \epsilon_{u,mn}. \qquad (2.35)
\end{aligned}$$

These recursions are initialized with $\beta_v(n) = f_{v,n}(w_v)$ for all v in the smallest scale. Then, these values are used to compute the initial values for $\beta_{\rho(v),v}(n)$ for that smallest scale an these are then used to compute $\beta_v(n)$ for the upper scale. The procedure is repeated until reaching the coarsest scale at the root node of the tree.

Similarly, $\alpha_u(n)$ can be computed with the recursion

$$\begin{aligned}
\alpha_u(n) &= \mathrm{p}\left(\mathcal{T}_{1 \setminus u}, r_u = n \left| \theta\right.\right) = \\
&= \sum_{m=1}^{M} \mathrm{p}\left(r_u = n, r_{\rho(u)} = m, \mathcal{T}_{1 \setminus \rho(u)}, \mathcal{T}_{\rho(u) \setminus u} | \theta\right) \\
&= \sum_{m=1}^{M} \mathrm{p}\left(r_u = n | r_{\rho(u)} = m, \theta\right) \frac{\mathrm{p}\left(\mathcal{T}_{\rho(u)} | r_{\rho(u)} = m, \theta\right)}{\mathrm{p}\left(\mathcal{T}_u | r_{\rho(u)} = m, \theta\right)} \cdot \\
&\qquad \cdot \mathrm{p}\left(\mathcal{T}_{1 \setminus \rho(u)}, r_{\rho(u)} = m | \theta\right) \\
&= \sum_{m=1}^{M} \frac{\epsilon_{u,nm} \beta_{\rho(u)}(m) \alpha_{\rho(u)}(m)}{\beta_{\rho(u),u}(m)}. \qquad (2.36)
\end{aligned}$$

This recursion is initialized with $\alpha_1(m) = \mathrm{p}(r_1 = m|\theta) = \kappa_m$. Note that using these variables, the likelihood of the model can be computed efficiently as

$$\begin{aligned}
\mathrm{p}(\mathbf{w}|\theta) &= \mathrm{p}(\mathcal{T}_1|\theta) \\
&= \sum_{n=1}^{M} \alpha_u(n) \beta_u(n) \qquad (2.37)
\end{aligned}$$

Note also that this computation does not depend on the node u chosen for splitting the tree.

Using these variables, the E step of the EM algorithm for HMT reduces to compute

$$\begin{aligned}\xi_u^\ell(m,n) &= \frac{\beta_u(m)\epsilon_{u,mn}\alpha_{\rho(u)}(n)\beta_{\rho(u)}(n)/\beta_{\rho(u),u}(n)}{\sum_{n=1}^{M}\alpha_u(n)\beta_u(n)},\\ \gamma_u^\ell(m) &= \frac{\alpha_u(m)\beta_u(m)}{\sum_{n=1}^{M}\alpha_u(n)\beta_u(n)}.\end{aligned}$$

These quantities remain fixed in the M step to update the model parameters. The state-transition probabilities in the HMT model are estimated by maximizing

$$\mathcal{Q}_\epsilon = \sum_\ell\sum_u\sum_m\sum_n \xi_u^\ell(m,n)\log\epsilon_{u,mn},$$

with the constraint

$$\sum_{m=1}^{M}\epsilon_{u,mn} = 1. \tag{2.38}$$

We obtain [24, 34]

$$\epsilon_{u,mn} = \frac{\sum_{\ell=1}^{L}\xi_u^\ell(m,n)}{\sum_{\ell=1}^{L}\gamma_{\rho(u)}^\ell(n)}. \tag{2.39}$$

Assume now that we model each conditional density $f_{u,m}(w_u^\ell)$ with a normal distribution with parameters $\mu_{u,m}$ and $\sigma_{u,m}^2$. This is a scalar density $\mathrm{p}(w_u^\ell = w|r_u = m,\theta) = \mathcal{N}(w|\mu_{u,m},\sigma_{u,m}^2)$. Estimation of the set of parameters $\{\mu_{u,m},\sigma_{u,m}^2\}_{u,m,n}$ is carried out by maximizing the auxiliary function

$$\begin{aligned}\mathcal{Q}_f(\theta,\theta^{old})(\{\mathbf{w}_\ell\}) &= \sum_\ell\sum_u\sum_m \gamma_u^\ell(m)\log f_{u,m}(w_u^\ell)\\ &= -\frac{1}{2}\sum_\ell\sum_u\sum_m \gamma_u^\ell(m)\left\{\frac{(w_u^\ell-\mu_{u,m})^2}{\sigma_{u,m}^\ell}+B\right\},\end{aligned}$$

with $B = \log 2\pi + \log \sigma^2_{u,m}$. We obtain [24, 34]

$$\mu_{u,m} = \frac{\sum_{\ell=1}^{L} \gamma_u^\ell(m) w_u^\ell}{\sum_{\ell=1}^{L} \gamma_u^\ell(m)}, \tag{2.40}$$

$$\sigma^2_{u,m} = \frac{\sum_{\ell=1}^{L} \gamma_u^\ell(m)(w_u^\ell - \mu_{u,m})^2}{\sum_{\ell=1}^{L} \gamma_u^\ell(m)}. \tag{2.41}$$

Inference in the HMT. Like in the case of standard HMM, we are often interested in inferring about the most probable sequence of states in the nodes of the tree that has generated the observed set of wavelet coefficients; that is, given $\mathbf{w}$, we look for the sequence of states $\tilde{\mathbf{r}}$ so that

$$\tilde{\mathbf{r}} = \arg\max_{\mathbf{r}} \mathrm{p}(\mathbf{r}|\mathbf{w}, \theta). \tag{2.42}$$

The specific algorithm for the HMT was first introduced by [34], but is analogous to Viterbi's algorithm for HMM presented above. In particular, the algorithm turns out to be a modified upward recursion, where the summation in (2.35) is replaced by taking the maximum over the states. In this way, the algorithm starts by initializing the variables $\lambda_u(m) = \beta_u(m)$ in the nodes u that corresponds to leaves of the tree. From this point, the following quantities are computed upwards the tree for each scale

$$\lambda_{\rho(u),u}(n) = \max_{1 \le m \le M} \beta_u(m) \epsilon_{u,mn}, \tag{2.43}$$

$$\xi_u(n) = \arg\max_{1 \le m \le M} \beta_u(m) \epsilon_{u,mn}, \tag{2.44}$$

$$\lambda_u(m) = f_{u,m}(w_u) \prod_{v \in C(u)} \lambda_{\rho(u),v}(m). \tag{2.45}$$

The recursion ends at the root node of the tree. Then, starting with

$$\tilde{r}_1 = \arg\max_{1 \le m \le M} \lambda_1(m),$$

for $u = 2, 3, \ldots, N$ we do

$$\tilde{r}_u = \xi_u(\tilde{r}_{\rho(u)}). \tag{2.46}$$

Limitations. In last years the HMT model has received considerable attention for several applications, including signal processing [32, 43, 84], image denoising [58, 59, 78, 85], texture classification [74, 82], computer vision [38, 95] and writer identification [46]. For classification tasks, however, it can deal only with static patterns. This limitation arises from the use of the discrete wavelet transform (DWT), which makes the structure of representations depend on the size of signals or images. To overcome this we could think of tying parameters along scales, but it would come at the price of reducing modeling power. In a typical scenario for pattern recognition we have multiple observations available and we would like to use the whole information in order to train a full model. In these cases, the HMT should be trained and used only with signals or images with the same size; otherwise, a warping preprocessing would be required to match different sizes and that would be difficult to achieve on-line.

2.4.3 Dealing with sequential data: the HMM-HMT model

A composite Markov model in the wavelet domain was introduced by Milone et al. [73] to deal with length variability in the observed sequences. The approach exploit the probabilistic nature of the HMT to embed it as the observation model for a standard HMM. An adapted version of the EM algorithm was derived to drive the parameter estimation of fully coupled models. The resulting structure is a composite hidden Markov model in which the HMT accounts for local features in a multiresolution framework while the external HMM handles dependencies in a larger time scale and adds flexibility to deal with sequential data. With this model, signals are seen as realizations of a random process which emits wavelet coefficients in a short term basis driven by a Markov chain. The emitted coefficients are not independent, but obey probabilistic dependencies structured as a tree.

To clarify, let us briefly describe this composite model. Let $\mathbf{w}^t \in \mathbb{R}^N$ be the set of coefficients emitted at time t and $\mathbf{W} = \{\mathbf{w}^1, \ldots, \mathbf{w}^T\}$ be the entire sequence of vectors of coefficients resulting from the DWT analysis. The observation is modeled by a HMM with a structure as defined in Section 2.1. In the assumed

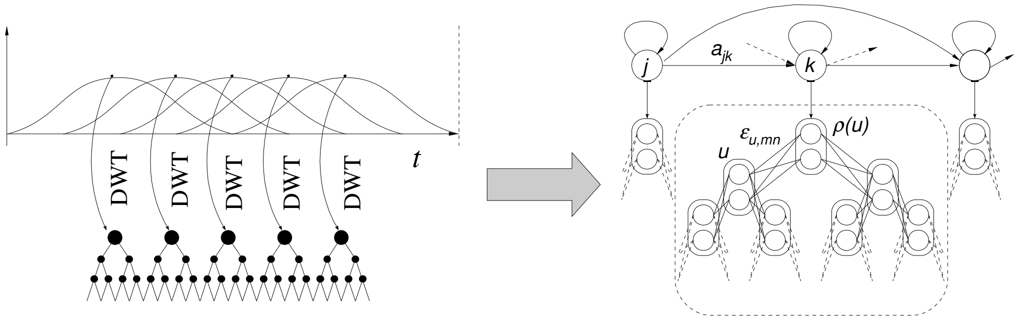


Figure 2.6: *The HMM-HMT model. A left-to-right hidden Markov model uses hidden Markov trees as models for the observed data in the wavelet domain.*

model, for every state k of the chain, observed coefficients are drawn from a HMT, so that $b_k(\mathbf{w}^t)$ is itself a hidden Markov structure. Figure 2.6 shows a sketch of the full model.

We recall that the observed coefficients w_u^t are drawn from an observation model $f_{u,m}(w_u^t)$ conditioned on the state m of the node. We assume *scalar* Gaussian models $\mathcal{N}(w_u^t|\mu_{u,m}, \sigma_{u,m}^2)$ for all of them. Finally, we will use superscript k to indicate the parameters of the HMT model θ^k that serves as observation model $b_k(\mathbf{w}^t)$ for the HMM.

Model likelihood and parameter estimation. Replacing (2.28) in (2.5), the likelihood for the composite HMM-HMT model given a single observed sequence $\mathbf{W}$ is:

$$\begin{aligned}
\mathcal{L}_\vartheta(\mathbf{W}) &= \sum_{\forall \mathbf{q}} \prod_t a_{q^{t-1}q^t} b_{q^t}(\mathbf{w}^t) \\
&= \sum_{\forall \mathbf{q}} \prod_t a_{q^{t-1}q^t} \sum_{\forall \mathbf{r}} \prod_{\forall u} \epsilon^{q^t}_{u,r_u^t r_{\rho(u)}^t} f^{q^t}_{u,r_u^t}(w_u^t) \\
&= \sum_{\forall \mathbf{q}} \sum_{\forall \mathbf{R}} \prod_t a_{q^{t-1}q^t} \prod_{\forall u} \epsilon^{q^t}_{u,r_u^t r_{\rho(u)}^t} f^{q^t}_{u,r_u^t}(w_u^t),
\end{aligned} \tag{2.47}$$

where we have assumed a left-to-right HMM. In these expressions, $\forall \mathbf{q}$ denotes that the sum is over all possible state sequences $\mathbf{q} = q^1, q^2, \ldots, q^T$ in the external HMM and $\forall \mathbf{R}$ accounts for all possible sequences of all possible combinations of hidden states $\mathbf{r}^1, \mathbf{r}^2, \ldots, \mathbf{r}^T$ in the nodes of each tree.

Parameters in the HMM-HMT model are estimated using an adapted version of the EM algorithm [73, 72]. The re-estimation formulas turn to be extensions of those stated previously for the HMT and HMM. We present the final results here, further details can be found in [73]. Assume we have P independent training sequences in the learning set, each with a number T_p of correlated observations. We have

- State-transition probabilities in the HMTs:

$$\epsilon^k_{u,mn} = \frac{\displaystyle\sum_{p=1}^{P}\sum_{t=1}^{T_p} \gamma^{p,t}(k)\xi^{p,tk}_u(m,n)}{\displaystyle\sum_{p=1}^{P}\sum_{t=1}^{T_p} \gamma^{p,t}(k)\gamma^{p,tk}_{\rho(u)}(n)}. \tag{2.48}$$

- Means of the conditional normal models in the HMTs:

$$\mu^k_{u,m} = \frac{\displaystyle\sum_{p=1}^{P}\sum_{t=1}^{T_p} \gamma^{p,t}(k)\gamma^{p,tk}_u(m)w^{p,t}_u}{\displaystyle\sum_{p=1}^{P}\sum_{t=1}^{T_p} \gamma^{p,t}(k)\gamma^{p,tk}_u(m)}. \tag{2.49}$$

- Variances of the conditional normal models in the HMTs

$$(\sigma^k_{u,m})^2 = \frac{\displaystyle\sum_{p=1}^{P}\sum_{t=1}^{T_p} \gamma^{p,t}(k)\gamma^{p,tk}_u(m)\left(w^{p,t}_u - \mu^k_{u,m}\right)^2}{\displaystyle\sum_{p=1}^{P}\sum_{t=1}^{T_p} \gamma^{p,t}(k)\gamma^{p,tk}_u(m)}. \tag{2.50}$$

where $\gamma^{p,tk}_u(m)$ and $\xi^{p,tk}_u(m,n)$ are computed as described for general HMM.

2.5 Concluding remarks

In this chapter we have reviewed the basics of HMM and have described briefly these models with two types of observation densities: Gaussian distributions and HMTs. Likelihood computation, parameter estimation and inference have been discussed for both of these models. Parameter estimation for Gaussian HMM will be revisited when we discuss sufficient dimension reduction methods for hidden Markov models. On the other hand, learning parameters of HMM-HMT models under maximum likelihood estimation will provide us the initial values for the iterative discriminative training procedure we develope in Chapter 3.

CHAPTER 3

Discriminative training of HMM in the wavelet domain

3.1 Introduction

Discriminative training of HMM has been a topic of intense research in recent years [45, 44, 52]. HMM-based classifiers designed in this way have shown to outperform their ML-based counterparts in many applications [13]. Most of these works deal only with standard HMM with Gaussian densities as observation models [13, 54, 1]. On the other hand, the HMM-HMT reviewed in Section 2.4.3 achieved promising results both for pattern recognition and for denoising tasks [72, 73]. Nevertheless, training algorithms used so far provide ML estimates for the parameters of this model.

The goal of this chapter is to take the MCE learning approach to this different scenario in which data is observed in the wavelet-domain and modeled through the HMM-HMT, aiming at improving the performance of these models for classification tasks.

3.2 MCE approach for classifier design

The classification rule $Y = f(\mathbf{W})$ usually depends on a parameterized set of functions or models, one for each class, which measure the degree of membership of the observation $\mathbf{W}$ to that class. Let $\{g_j(\mathbf{W};\Theta)\}_{j=1}^{h}$ be that parameterized set of functions for a classification task comprising h classes $c_1, c_2, \ldots, c_h$, and $\Theta = \{\vartheta_j\}_{j=1}^{\mathcal{M}}$ be the whole parameter set. An unlabeled observation $\mathbf{W}$ will be assigned to class c_i when

$$f(\mathbf{W};\Theta) \triangleq \arg\max_j \{g_j(\mathbf{W};\Theta)\} = i \; . \tag{3.1}$$

The classifier design involves the estimation of an optimum parameter set Θ^* that minimizes the expected classification error over all the observation space.

In traditional generative learning, $g_j(\mathbf{W};\Theta)$ is set to the joint distribution of $(\vartheta_j, \mathbf{W})$ and maximizing (3.1) amounts to maximizing $\mathrm{p}(\vartheta_j|\mathbf{W})$. Then, by the Bayes rule, the model for each class can be trained by maximizing the likelihood $\mathrm{p}(\mathbf{W}|\vartheta_j)$ using a training sample from class c_j only. On the other hand, in discriminative learning all models are updated simultaneously in a competitive way. This process aims to exploit differences between classes that can lead to a reduction in the error rate of the classifier. In MCE training in particular, minimization of the classification error is set formally as a goal. We now summarize the main topics of the method and provide simulation examples with a simple Gaussian model in order to motivate our developments.

3.2.1 Derivation of the MCE criterion

The main ingredient of the MCE approach for classifier design is a soft approximation of the misclassification risk over the set of samples available for training. Although in advance we would not be able to guarantee minimum expected error over all possible observations working just on a finite (possibly small) training set, the method has shown to generalize well over validation sets [69, 92]. Recent works have also explained the generalization property of MCE methods by linking them with large margin estimation [52, 70].

For an observation $\mathbf{W}$, the conditional risk of misclassification is given by

$$\mathcal{R}(\Theta|\mathbf{W}) = \sum_{j=1}^{\mathcal{M}} \ell(f(\mathbf{W};\Theta), c_j)\mathrm{P}(c_j|\mathbf{W}),$$

where $\ell(f(\mathbf{W};\Theta), c_j)$ is a loss function which penalizes a wrong decision when classifying an observation $\mathbf{W}$ from class c_j. The usual choice for the loss function is the zero-one loss which assigns $\ell(f(\mathbf{W}), c_j;\Theta) = 1$ for $f(\mathbf{W}) \neq c_j$ and zero for correct classification [33]. In the training process, we look for a parameter set Θ^* that minimizes the risk

$$\mathcal{R}(\Theta) = \int \sum_{j=1}^{\mathcal{M}} \ell(f(\mathbf{W};\Theta), c_j)\mathrm{P}(c_j|\mathbf{W})d\mathrm{P}(\mathbf{W}), \tag{3.2}$$

where the integral extends over the entire sequence space. Nevertheless, when designing a classifier we only have the labeled observations in the training set. Let Ω_j stand for the subset of observations in the training set which belong to class c_j. The expectation (3.2) can be replaced with an average of the loss with all the observations given equal probability mass

$$\tilde{\mathcal{R}}(\Theta) = \frac{1}{S}\sum_{s=1}^{S}\sum_{j=1}^{h} \ell(f(\mathbf{W}_s;\Theta), c_j)\mathcal{I}(\mathbf{W}_s \in \Omega_j).$$

In the equation above $\mathcal{I}(\cdot)$ is the indicator function and S is the cardinality of the training set.

The MCE approach minimizes a smoothed version of this empirical risk which is differentiable with respect to model parameters [54]. Let us write this approximation as $\ell(f(\mathbf{W};\Theta), c_j) = \ell(d_j(\mathbf{W};\Theta))$, where function $d_j(\mathbf{W};\Theta)$ simulates the decision of the classifier. Assume the current training observation comes from class c_i. A common choice for $\ell(d_i(\mathbf{W};\Theta))$ is the sigmoid [13, 54]

$$\ell(d_i(\mathbf{W};\Theta)) = \frac{1}{1+\exp\left(-\gamma d_i(\mathbf{W};\Theta)+\beta\right)} \ . \tag{3.3}$$

Parameter γ controls the sharpness of the sigmoid and the bias β is usually set to zero. To complete the picture we must specify the function $d_i(\mathbf{W};\Theta)$, which is often referred to as the *misclassification function* [13, 54, 55]. In order to allow $\ell(d_i(\mathbf{W};\Theta))$ to behave close to the zero-one loss, it must give a large enough positive value for strongly misclassified observations and a small negative value when the decision is right. In addition, very confusing samples should give a value close to zero so that their related loss fall in the raising segment of the sigmoid. From (3.1), an obvious candidate for $d_i(\mathbf{W};\Theta)$ is

$$d_i(\mathbf{W};\Theta) = \max_{j \neq i}\{g_j(\mathbf{W};\Theta)\} - g_i(\mathbf{W};\Theta) \ .$$

However, the maximum operation is not differentiable. As we are looking for a smoothed version of the risk, what is used in practice is a soft approximation like an ℓ_p-norm with p large. However, different selections of the misclassification function are possible (see, for example, [55]) and they can have important effects on the performance of the algorithm as we will see below.

3.2.2 Optimization

In the preceding section we have described the approximation of the empirical risk which serves as the optimization criterion for MCE learning. The simplest approach to find the parameter estimates is a gradient-based optimization technique often known as Generalized Probabilistic Descent (GPD), which is a special case of stochastic approximation [13, 14, 55]. This is simply an on-line scheme which aims at minimizing the smoothed approximation of the classification risk by updating the whole set of parameters Θ in the steepest-descent direction of the loss. Starting from an initial estimate $\hat{\Theta}_0$, the τ-th iteration of the algorithm can be summarized as

$$\hat{\Theta} \longleftarrow \hat{\Theta} - \alpha_\tau \ \nabla_\Theta \ell(\mathbf{W}_\tau;\Theta)\big|_{\Theta=\hat{\Theta}_\tau} \ , \tag{3.4}$$

where α_τ is the learning rate, that is allowed to decrease gradually as iterations proceed in order to assure convergence [55]. Usually, $\hat{\Theta}_0$ is chosen to be the ML estimate of Θ and the updating process is carried out for each training signal [13], so that $\mathbf{W}_\tau$ is actually the sequence picked up from the training set at the τ-th iteration. Batch implementations can also be used to exploit parallelization [52, 69]. It is important to see that the derivative of (3.3) on $d_i(\mathbf{W},\Theta)$ is symmetric around zero when $\beta = 0$. As a consequence, the strength of the update depends on how confusing the training observation is to the classifier and not on the

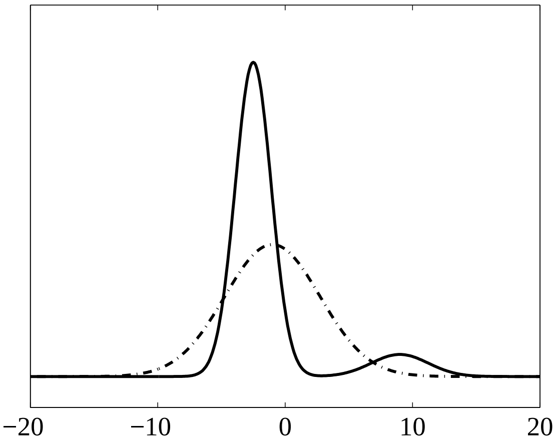


Figure 3.1: *Distribution of the data for the proposed experiment. The solid line shows the distribution of class A while the dotted line shows the one of class B.*

correctness of the decision. This way, patterns that are similarly likely to belong to different classes induce the update of the parameter set, even if they are well classified.

3.2.3 An example with Gaussian models

In order to show the potential of discriminative learning over traditional ML estimation of model parameters, let us consider a simulation example for a binary classification problem. We assume Gaussian models for both classes, but allow data from one of them, say class A, to be drawn actually from a two-component Gaussian mixture, with parameters $\mu_{A1} = -2.5$, $\sigma^2_{A1} = 4$, $\mu_{A2} = 9$, $\sigma^2_{A2} = 9$ and weights 0.9 and 0.1, respectively. This is a simple example of a model not fitting the real distribution of observed data. To make the decision task more difficult, suppose also that the real distribution of class B data is a Gaussian with mean and variance very close to the global mean and variance for class A. Figure 3.1 illustrates the proposed situation. It is clear that this is a very demanding task for a quadratic classifier based on ML estimation. In fact, we expect it to discriminate very poorly and we are interested in seeing how much improvement can the MCE approach achieve.

Ten runs were carried out for each training method. For every run, data was

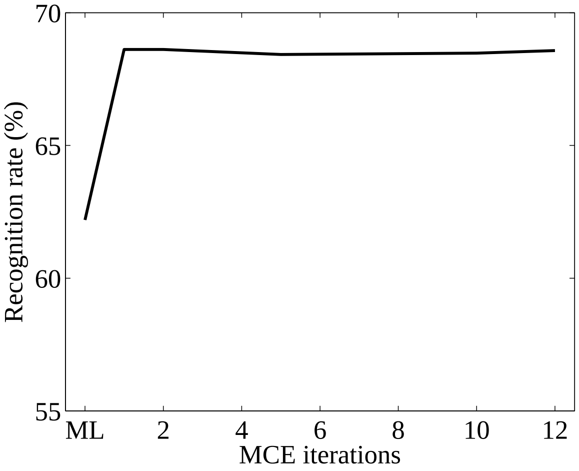


Figure 3.2: *Recognition rates over the testing set as a function of the number of MCE iterations. Shown scores are averages over ten runs for each tested condition.*

generated randomly for class A first and its sample mean and variance were used to generate data from class B, setting $\mu_B = \hat{\mu}_A + 0.25$ and $\sigma_B^2 = \widehat{\sigma}_A^2$. A thousand samples from each class were used in both the training set and a separate testing set. ML estimates were used as initial guesses for the discriminative training, and standard settings were used for the MCE criterion [1]. Obtained results varying the number of MCE iterations are shown in Figure 3.2. It can be seen that an important improvement in recognition rate is achieved after just a few iterations of the algorithm. After five iterations, the discriminative approach reduces the error rate from 38% to 31%. Further iterations do not seem to provide significant improvements for this case.

Figure 3.3 compares the trained models obtained with maximum likelihood only against those estimated discriminatively. The competitive updating process modifies initial model parameters so that the Gaussian for class A concentrates around the mean for the most likely component in the original mixture. On the other hand, the model for class B widens a lot to account for all other values in data. The final models used for classification are very different from the real data distributions. Thus, unlike with the ML approach, obtained parameter estimates do not try to explain the data but only to improve the classifier performance emphasizing differences between distributions.

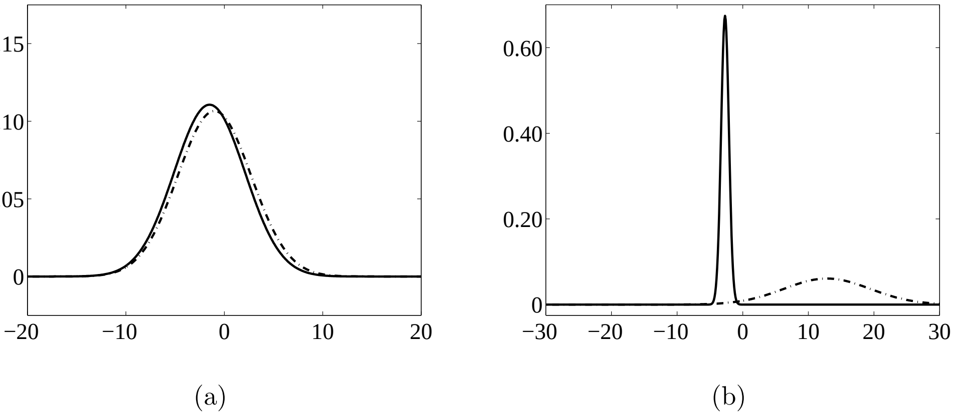


Figure 3.3: *Comparison of the trained classifiers, showing the densities they use for classification. a) Models obtained with maximum likelihood estimation. b) Models obtained with MCE training after five iterations over the whole training set. Solid lines show the model for class A and dotted lines show the one for class B.*

3.3 Algorithm formulation

It is clear from our discussion of the general aspects of the MCE/GPD approach in Section 3.2 that the key points to be defined when designing a classifier under this framework are: i) the parametrized form for the discriminant functions; and ii) the misclassification function $d_i(\mathbf{W};\Theta)$. If an unconstrained optimization algorithm like GPD is to be used, suitable transformations of the parameters must also be introduced to account for constraints. We will follow rather conventional choices for i) and for transformation of parameters in Section 3.3.1, but we will go apart from the mainstream when considering ii) in Section 3.3.2. Updating formulas are outlined in Section 3.3.3, while details about their derivation are left to Appendix A.

3.3.1 Discriminant functions and parameter transformations

For an HMM-based discriminant function approach to pattern recognition, it is a usual practice to define $g_j(\mathbf{W};\Theta)$ as a function of the joint likelihood $\mathcal{L}_{\vartheta_j}$ [13]. In particular, due to the efficiency of Viterbi's decoding algorithm for both

HMM and HMT, it is attractive to define

$$\begin{aligned} g_j(\mathbf{W};\Theta) &= \left|\log\left(\max_{\mathbf{q},\mathbf{R}}\left\{\mathcal{L}_{\vartheta_j}(\mathbf{W},\mathbf{q},\mathbf{R})\right\}\right)\right| \qquad (3.5)\\ &= -\sum_t \log a_{\bar{q}^{t-1}\bar{q}^t} - \sum_t\sum_{\forall u}\log \epsilon^{\bar{q}^t}_{u,\bar{r}^t_u\bar{r}^t_{\rho(u)}} - \sum_t\sum_{\forall u}\log f^{\bar{q}^t}_{u,\bar{r}^t_u}(w^t_u)\ , \end{aligned}$$

where $|\cdot|$ denotes absolute value and $\bar{q}^t$ and $\bar{r}^t$ refer to states in the external HMM and the corresponding HMT model, respectively, that achieve maximum joint likelihood. It should be noticed that this definition involves a little change in what we have said about the decision of the classifier in (3.1). Now this decision is ruled by the minimum (rather than the maximum) of the discriminant functions, valued at the unlabeled observation.

Despite of discriminant functions using standard model parameters, we must introduce some parameter transformations to account for restrictions if we are to use a gradient-based optimization technique such as GPD [13, 54]. To constrain a_{ij} to be a probability, we define $\tilde{a}_{ij}$ so that

$$a_{sj} = \frac{\exp\tilde{a}_{sj}}{\sum_m \exp\tilde{a}_{sm}}. \qquad (3.6)$$

Exponentiation assures a_{ij} is non-negative and normalization makes it less or equal to one. A similar transformation is needed for the transition probabilities in the internal HMTs. With analogous arguments, we define $\tilde{\epsilon}^k_{u,mn}$ so that

$$\epsilon^k_{u,mn} = \frac{\exp\tilde{\epsilon}^k_{u,mn}}{\sum_p \exp\tilde{\epsilon}^k_{u,pn}}. \qquad (3.7)$$

We also need to constrain the Gaussian variances to be positive-valued. To do so, we define $\tilde{\sigma}^k_{u,m}$ so that $\tilde{\sigma}^k_{u,m} = \log\sigma^k_{u,m}$. In addition, we scale the means of the Gaussian distributions as $\tilde{\mu}^k_{u,m} = \mu^k_{u,m}/\sigma^k_{u,m}$. This is done to reduce the range of values that the parameters can take, so that the same learning rate can be used for all of them [54]. Note that these transformations are rather standard in the literature [13, 54].

3.3.2 Misclassification function

For HMMs with Gaussian mixture observations and discriminant functions defined as the negative of those stated above, the frequent choice for MCE training has been simulating the decision of the classifier with the function [13]

$$\tilde{d}_i(\mathbf{W};\Theta) = -\tilde{g}_i(\mathbf{W};\Theta) + \log\left[\frac{1}{h-1}\sum_{j\neq i} e^{\tilde{g}_j(\mathbf{W};\Theta)\eta}\right]^{1/\eta} . \tag{3.8}$$

As η becomes arbitrarily large the term in brackets approximates, up to a constant, the supremum of $\{\tilde{g}_j(\mathbf{W};\Theta)\}$ for all $j \neq i$. This definition of the misclassification function, composed with a zero-bias approximation to the zero-one loss, penalizes confussing patterns rather than a wrong classifcation. Thus, a strong decision of the classifier implies no update of the parameter set, whether this decision is right or not. Despite it can look counterintuitive at first, it is in fact a conservative statement which avoids modifying parameter estimates due to bad data.

Nevertheless, likelihoods for the HMT model are tipically much smaller than those found for Gaussian mixtures in standard feature spaces. We can expect this noting that the joint likelihood for the HMM-HMT model involves many products which are probabilities often being very small. As a result, $g_j(\mathbf{W};\Theta)$ takes extremely low values for $\mathbf{W} \notin \Omega_j$ and the exponentiation leads to numerical underflow. A natural option to look for a similar behaviour of the misclassification function but avoiding those numerical issues is to define it as

$$\bar{d}_i(\mathbf{W};\Theta) = g_i(\mathbf{W};\Theta) - \left[\frac{1}{h-1}\sum_{j\neq i} g_j(\mathbf{W};\Theta)^{-\eta}\right]^{-1/\eta} . \tag{3.9}$$

Roughly speaking, both of these functions account for the decision margin between the true model and the best competing ones. They weight rival candidates, but do not introduce any special corrective penalty in case of a wrong classification. Because of this, we will refer to them as symmetric misclassification functions and will use the acronym SMF to refer to (3.9) in what follows.

Due to the behaviour of the likelihoods for the HMM-HMT model discussed above, also their dispersion is much larger than in the Gaussian mixture-HMM case. In this situation, similarity could be better measured comparing the order of magnitude between discriminant functions rather than their difference. To do so, we define an alternative form for discriminant functions as

$$d_i(\mathbf{W};\Theta) = 1 - \frac{\left[\frac{1}{h-1}\sum_{j\neq i} g_j(\mathbf{W};\Theta)^{-\eta}\right]^{-1/\eta}}{g_i(\mathbf{W};\Theta)} \ . \tag{3.10}$$

As above, η is supposed to be a large positive scalar so that the sum in the numerator approaches the minimum of the terms as η grows. When the classifier takes a right decision, this minimum will be larger than $g_i(\mathbf{W};\Theta)$ and $d_i(\mathbf{W};\Theta)$ will take a negative value as required. If the observation makes decision hard for the classifier, $d_i(\mathbf{W};\Theta)$ will be close to zero. However, it must be noticed that $d_i(\mathbf{W};\Theta)$ will take no value larger than one. This implies that all misclassified observations will fall in the raising segment of the approximation to the zero-one loss if it is not too sharp. This simple fact has a very important effect in practice because it determines that every misclassified observation in the training set induces an update of the parameter set. To stress this lack of symmetry in dealing with correct and wrong classifications, we will refer to (3.10) as a no-symmetric misclassification function and will use the acronym nSMF to denote it in the following.

3.3.3 Updating formulas

In the following, let assume that the τ-th training sequence $\mathbf{W}_\tau$ belongs to Ω_i. To simplify notation, allow ℓ_i, d_j and g_j stand for $\ell_i(d_j(\mathbf{W};\Theta))$, $d_j(\mathbf{W};\Theta)$ and $g_j(\mathbf{W};\Theta)$, respectively. For convenience, define also

$$\zeta_{ii} \triangleq \frac{\mathrm{d}\ell_i}{\mathrm{d}d_i}\frac{\partial d_i}{\partial g_i} \ ,$$

and

$$\zeta_{ij} \triangleq \frac{\mathrm{d}\ell_i}{\mathrm{d}d_i}\frac{\partial d_i}{\partial g_j} \ ,$$

where in the last expression we assume $i \neq j$. For the misclassification function SMF, these quantities take values

$$\begin{aligned} \zeta_{ii} &= \gamma \ell_i (1-\ell_i) & (3.11) \\ \zeta_{ij} &= \gamma \ell_i (1-\ell_i)(d_i - g_i) \frac{g_j^{-\eta-1}}{\sum_{k \neq i} g_k^{-\eta}} \ . & (3.12) \end{aligned}$$

Note that for a binary classification problem these quantities have the same absolute value but opposite sign. For the misclassification function nSMF, we have

$$\begin{aligned} \zeta_{ii} &= \gamma \ell_i (1-\ell_i) \frac{d_i - 1}{g_i} & (3.13) \\ \zeta_{ij} &= \gamma \ell_i (1-\ell_i)(1 - d_i) \frac{g_j^{-\eta-1}}{\sum_{k \neq i} g_k^{-\eta}}. & (3.14) \end{aligned}$$

Again, ζ_{ii} and ζ_{ij} always have opposite sign, but their absolute value is not the same even for a two-classes only task.

The updating process works upon the transformed parameters to assure the original ones remain in their feasibility range. For the Gaussian mean associated to the state m in the node u of the HMT linked to the state k of the HMM for class c_j, the updating step is given by

$$\tilde{\mu}_{u,m}^{(j)k} \longleftarrow \tilde{\mu}_{u,m}^{(j)k} - \alpha_\tau \left. \frac{\partial \ell_i}{\partial \tilde{\mu}_{u,m}^{(j)k}} \right|_{\Theta = \hat{\Theta}_\tau} , \qquad (3.15)$$

where $\hat{\Theta}_\tau$ refers to the estimates of parameters obtained in the previous iteration. Applying the chain rule of differentiation and using the variables defined above, we get (see details in Appendix A):

$$\tilde{\mu}_{u,m}^{(j)k} \longleftarrow \tilde{\mu}_{u,m}^{(j)k} - \alpha_\tau \zeta \sum_t \delta(\bar{q}^t - k, \bar{r}_u^t - m) \left[\frac{w_u^t - \hat{\mu}_{u,m}^{(j)k}}{\hat{\sigma}_{u,m}^{(j)k}} \right] , \qquad (3.16)$$

where ζ takes the value ζ_{ii} or ζ_{ij} depending on whether we are dealing with a training pattern from the same class as the model or not. The delta function $\delta(\cdot,\cdot)$ is typical of Viterbi decoding. As the factor in brackets depends on the time frame through w_u^t, this function states that we only consider for the updating process the standardized observed coefficient for the node in those frames when the most likely state in the external model is k and the most likely state in the node is

m. Then, to restore the original parameters we just compute $\mu^{(j)k}_{u,m}(\tau+1) = \sigma^{(j)k}_{u,m}(\tau)\tilde{\mu}^{(j)k}_{u,m}(\tau+1)$. The updating process for Gaussian variances is completely analogous to the one shown above for the means. The working expression for training reads:

$$\tilde{\sigma}^{(j)k}_{u,m} \longleftarrow \tilde{\sigma}^{(j)k}_{u,m} - \alpha_\tau \zeta \sum_t \delta(\bar{q}^t - k, \bar{r}^t_u - m) \left[\left(\frac{w^t_u - \hat{\mu}^{(j)k}_{u,m}}{\hat{\sigma}^{(j)k}_{u,m}} \right)^2 - 1 \right] , \qquad (3.17)$$

where ζ and $\delta(\cdot,\cdot)$ have the same meaning as above. Once again, Viterbi decoding acting on the Markovian dependencies decouples all the nodes and the final formula resembles just the derivative of a log-normal on its standard deviation. Then, original variances are restored doing $\sigma^{(j)k}_{u,m}(\tau+1) = \exp(\tilde{\sigma}^{(j)k}_{u,m}(\tau+1))$.

The above strategy works for updating the transition probabilities too. It is shown in Appendix A that the updating formula for the transformed probability $\tilde{\epsilon}^{(j)k}_{u,mn}$ reads:

$$\begin{aligned} \tilde{\epsilon}^{(j)k}_{u,mn} \longleftarrow & \tilde{\epsilon}^{(j)k}_{u,mn} - \alpha_\tau \zeta \left\{ \sum_t \delta(\bar{q}^t - k, \bar{r}^t_u - m, \bar{r}^t_{\rho(u)} - n) - \right. \\ & \left. - \sum_t \sum_p \delta(\bar{q}^t - k, \bar{r}^t_u - p, \bar{r}^t_{\rho(u)} - n) \hat{\epsilon}^{(j)k}_{u,mn} \right\} . \end{aligned} \qquad (3.18)$$

The first sum in brackets counts how many times the most likely state in the node is m given that the most likely state in its parent node is n and the state in the HMM is most likely to be k. For the double sum, note that $\hat{\epsilon}^{(i)k}_{u,mn}$ is a common factor and the sum actually counts all the frames when the most likely state in the parent of the given node is n and the most likely state in the external HMM is that related to the corresponding HMT, k in this case. Restoration of the original parameters is straightforward from the definition of $\tilde{\epsilon}^{(j)k}_{u,mn}$.

Finally, following identical procedures we find the updating formulas for the transformed state transition probabilities $\tilde{a}^{(j)}_{sj}$ given by:

$$\tilde{a}^{(i)}_{sj} \longleftarrow \tilde{a}^{(i)}_{sj} - \alpha_\tau \zeta \left\{ \sum_{t=1}^{T} \delta(\bar{q}_{t-1} - s, \bar{q}_t - j) - \sum_{t=1}^{T} \delta(\bar{q}_{t-1} - s) \hat{a}^{(i)}_{sj} \right\} . \qquad (3.19)$$

Once again, we can interprete the summations in the above formula as counters acting on the sequence of most likely states in the external HMM, as given by Viterbi decoding. Original parameters $a_{sj}^{(j)}(\tau+1)$ are easily restored using the definition of $\tilde{a}_{sj}^{(j)}$.

3.4 Experimental results

In order to assess the proposed training method, we carry out automatic speech recognition tests using phonemes from the TIMIT database [99]. This is a well known corpus in the field and it has already been used in previous works dealing with similar schemes [71, 73]. In particular, we use samples of phonemes /b/, /d/, /eh/, /ih/ and /jh/. The voiced stops /b/ and /d/ have a very similar articulation and different phonetic variants according to the context. Vowels /eh/ and /ih/ were selected because their formants are very close [80]. Thus, these pairs of phonemes are very confusable. The affricate phoneme /jh/ was added as representative of the voiceless group to complete the set. It must be remarked that this signals are not spoken isolatedly but extracted from continuous speech. Because of that, there is a large variability in both acoustic features and duration in the dataset. All of these contribute to a very demanding task for a classifier.

As a measure of performance, we compare recognition rates achieved with the proposed method against those for the same models trained only using the EM algorithm. In all the experiments we model each phoneme with a left-to-right hidden Markov model with three states ($N_Q = 3$). The observation density for each state is given by an HMT with two states per node. This is the standard setting for the state space in most HMT applications [24]. The sequence analysis is performed on a short-term basis using Hamming windows 256-samples long, with 50% overlap between consecutive frames. On each frame, a full dyadic discrete wavelet decomposition is carried out using Daubechies wavelets with four vanishing moments [67, 73].

In a first set of experiments, we show numerically that the recognition rate achieved with the EM algorithm attains an upper bound for the given models and dataset. This bound is shown not to be surpassed neither increasing the number of re-estimations of the algorithm nor enlarging the training set. We next carry out a two-phoneme recognition task using the approach developed in Section 3.3. The re-estimation formulas are reduced to much simpler expressions in this case, allowing to get further insight into the discriminative training process. It also serves us to compare the misclassification functions proposed in Section 3.3.2. Finally, we carry out a multiclass speech recognition experiment to assess the error rate reduction after adding a discriminative stage to the training process.

3.4.1 Limits on performance for ML estimators

Discriminative training methods usually use ML estimates computed via the EM algorithm as initial values for model parameters [13, 52]. Thus, it is fair to ask if better performance could be achieved just using more training sequences in the pure ML approach or increasing the number of re-estimations in the EM algorithm, without adding a discriminative stage. To answer this question empirically for our data and our particular model, we first perform a two-phoneme recognition task using models trained with the EM algorithm proposed in [71, 73]. We ran the experiment using training sets of increasing sizes, from 25 sequences to 200. Each training set was picked at random from the whole training partition of the dataset. A separate testing set with 200 sequences was used for all trials. Each tested condition was run ten times and the number of re-estimations used for the EM algorithm was fixed at 6 in all of them. Obtained results for the {/b/,/d/} pair are given in Figure 3.4.a). It is clear from the figure that increasing the number of training samples does not lead to a significant improvement in the recognition rate when only the EM algorithm is used for training. In fact, the analysis of the results shows that the p-value for the {/b/,/d/} pair is 0.4476, which is far from the critical value to reject the null hypothesis of all means being statistically the same. Similar comments apply for the {/eh/,/ih/} pair.

On the other hand, the effect of fixing the size of the training set and increasing the number of re-estimations used in the EM algorithm is shown in Figure 3.4.b). Given values correspond to training sets with 200 sequences. It can be seen that recognition rates remain fairly the same with the increase in the number of re-estimations. For the {/b/,/d/} pair and the specific set of sequences used in the experiment, there is a small improvement in performance up to ten re-estimations. Beyond that there is no benefit in adding re-estimations steps in the EM algorithm. For the {/eh/,/ih/} pair of phonemes there is a little improvement

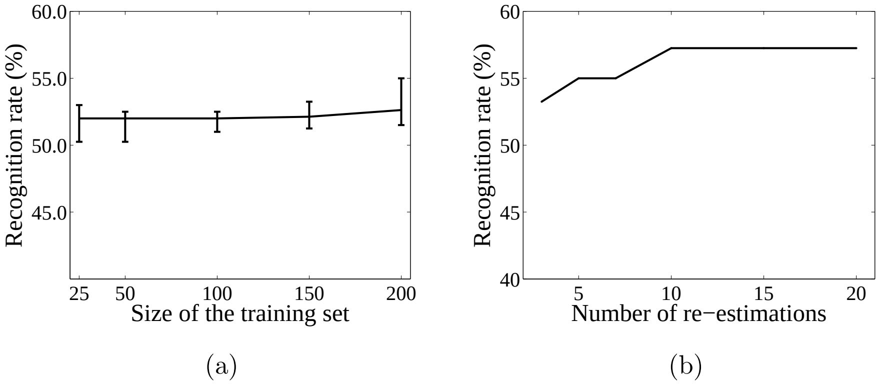


Figure 3.4: *Recognition rates for EM training. a) Increasing the size of the training set. Shown results are the median over ten runs for each tested condition. Error-bars are given by the first and third quartiles of the obtained scores. b) Increasing the number of reestimations. The $\{/b/,/d/\}$ pair was used in both experiments.*

up to five re-estimations but no further improvement is seen either adding more re-estimations.

Observed results in this experiment reproduce a typical scenario when working with “real” data. Always the proposed model it is obviously not the true model for the data in that case. Increasing the training set or adding re-estimations to the EM algorithm can only contribute to find better estimates for the parameters in those models. If models were the true ones, this would help for classification. But as models do not give the exact distribution of the data, we cannot expect this to translate into better discrimination. Note that this is not a statement on the goodness of fit of the model itself. For complex real data (like speech, in this case), hardly any model we propose would fail to model it accurately. Here is when discriminative training becomes important.

3.4.2 MCE training for two-class phoneme recognition

In order to get some insight into the learning process, we first consider a classification task comprising only two phonemes. In this case, for a training sequence $\mathbf{W} \in \Omega_1$, the symmetric misclassification function SMF in (3.9) reduces to

$$\bar{d}_1(\mathbf{W};\Theta) = g_1(\mathbf{W};\Theta) - g_2(\mathbf{W};\Theta) \ .$$

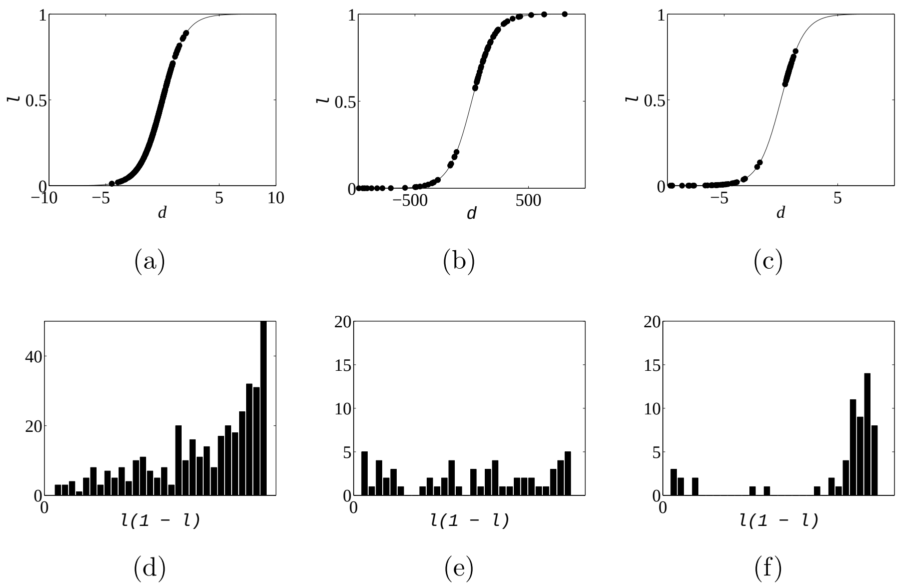
Aside from the change in sign to account for the different definition of the discriminant functions we made in (3.6), this is the same as the frequently used function (3.8) for a binary classification problem [1]. When the classifier decision is right, $g_1(\mathbf{W};\Theta) < g_2(\mathbf{W};\Theta)$ and the misclassification function takes a negative value. As this decision is stronger, $\bar{d}_1(\mathbf{W};\Theta)$ becomes more negative and the resulting loss (3.3) goes to zero. We then see from the updating formulas in Section 3.3.3 that no updating is performed in such a case. So, the algorithm preserves model parameters that do well when classifying the current training signal. Furthermore, for strongly confused patterns $\bar{d}_1(\mathbf{W};\Theta)$ becomes a large positive value and no update is introduced either.

On the other hand, the non-symmetric misclassification function nSMF in (3.10) reduces to

$$d_1(\mathbf{W};\Theta) = 1 - \frac{g_2(\mathbf{W};\Theta)}{g_1(\mathbf{W};\Theta)} \ .$$

When the classifier decision is right, it behaves closely to $\bar{d}_1(\mathbf{W};\Theta)$. Nevertheless, if the current training sequence is strongly misclassified, $d_1(\mathbf{W};\Theta)$ will tend to 1. Unlike the previous case, parameters will be updated unless γ is too large. Therefore, this definition of the misclassification function adds a corrective feature to the learning process. In both cases, the parameter update takes place when models are confusable and it is the strongest when the current training sequence is equally likely for both of them. With the second definition, however, we can also expect an updating step even for strongly misclassified patterns.

We can get an idea of the strength of the updating steps looking at the distribution of $\ell_i(1-\ell_i)$. For a given pattern, this factor scales the gradients in the re-estimation formulas according to how confusable the pattern is for the classifier, as told by the misclassification function. Figure 3.5 compares the distribution of this factor at the beginning of the iterative process, obtained for the same training set but choosing a different training method in each case. Figure 3.5.a) corresponds to standard MCE training for HMMs with Gaussian mixtures as observation densities on a cepstral-based feature space. Figure 3.5.b) comes from a classifier based on HMM-HMTs, using the misclassification function SMF to derive the MCE criterion; and Figure 3.5.c) comes from a classifier based on HMM-HMTs, but using nSMF as the misclassification function. In these later histograms, the bin that includes the value $\ell_i(1-\ell_i) = 0$ was removed to keep figures at a similar scale. It is interesting to see that despite of (3.8) and SMF sharing the same misclassification function for a binary problem like this, it is



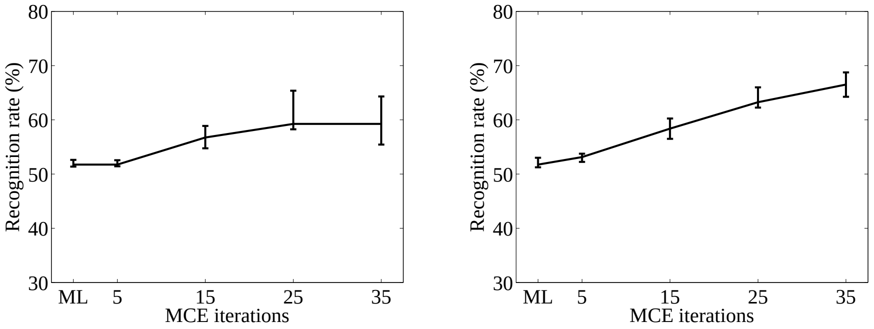
(a) (b) (c)

(d) (e) (f)

Figure 3.5: *Distribution of the loss and the factor $\ell_i(1 - \ell_i)$ at the beginning of different settings of the MCE training. Upper figures show the location of the loss for each sequence in the training set, while figures at the bottom show the resulting histogram for the factor $\ell_i(1 - \ell_i)$. a) and d) using cepstral features and Gaussian mixture-HMMs along with a standard misclassification function as in (3.8); b) and d) using the HMM-HMT and SMF; c) and e) using the HMM-HMT and nSMF.*

the criterion based on the misclassification function nSMF which generates the distribution of factors more similar to the standard case shown in plot d) when using the HMM-HMT. Therefore, changing the feature space used to represent the data can induce important modifications in the way the updating process is driven by a given approximation of the loss.

To compare the performance achieved by SMF and nSMF, we carried out numerical experiments with phonemes {/b/,/d/} and {/eh/,/ih/}, which are the most confused pairs in the set. Two hundred sequences from each class were used for training and another set of two hundred sequences from each class were used for testing. Five re-estimation steps were used in the EM algorithm, along with Viterbi flat start [80]. Parameters for the MCE learning stage were set following informal tests on a validation test, aimed to find the values that give better performance for each pair of phonemes and for each choice of misclassification function. When using SMF we set $\alpha_0 = 2.5$ and $\gamma = 0.01$, while we set $\alpha_0 = 0.5$ and $\gamma = 1$ for the algorithm derived using nSMF. In all cases, the learning rate was

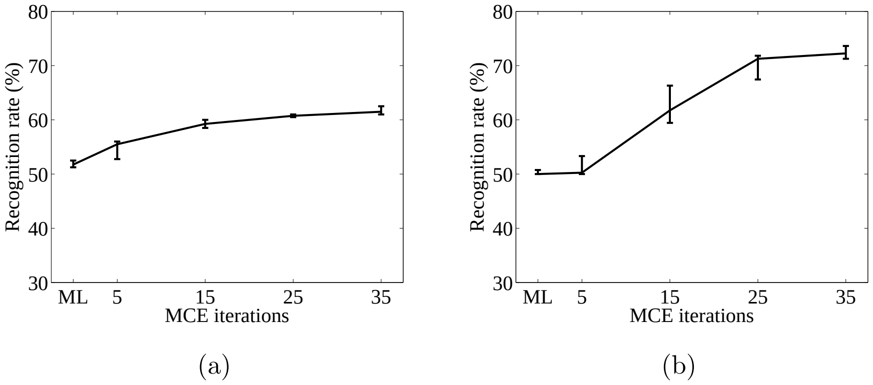


(a) (b)

Figure 3.6: *Recognition rates for phonemes /b/ and /d/: a) using SMF; b) using nSMF. Shown results are the median over ten runs for each tested condition. Error-bars are given by the first and third quartiles of the obtained scores.*

decreased at a constant rate from $\alpha_\tau = \alpha_0$ at the beginning of the discriminative training to $\alpha_\tau = 0$ at its end. The number of iterations of the MCE algorithm through the whole training set was varied as $5, 15, 25$ and 35. Ten runs were performed for each tested condition, varying the training set in each one but keeping fixed the set for testing.

Obtained results for each pair of phonemes and each choice of the misclassification function are shown in Figure 3.6 and Figure 3.7. Figure 3.6 shows the achieved recognition rates for the pair $\{$/b/,/d/$\}$. Performance for zero iterations of the MCE algorithm refers to the case when the classifier is trained using ML estimation and serves as the baseline for comparison. It can be seen that the scores using discriminative steps are significantly higher than the baseline with both MCE criteria for all tested conditions with more than five iterations. For five MCE iterations there is no significant improvement on the average. Figure also shows that the training method using the misclassification function nSMF outperforms that based on SMF. With 35 iterations of the algorithm, the former achieves an average reduction of about 30% in the error rate, whereas the later does a 14%. In addition, there seems to be a trend to continue rising the recognition rate in Figure 3.6.b), while in 3.6.a) improvements appear to have reached a bound. Furthermore, the variance of the obtained scores remain very similar as they go better for the method using the misclassification function nSMF, while it increases significantly for the method using SMF.

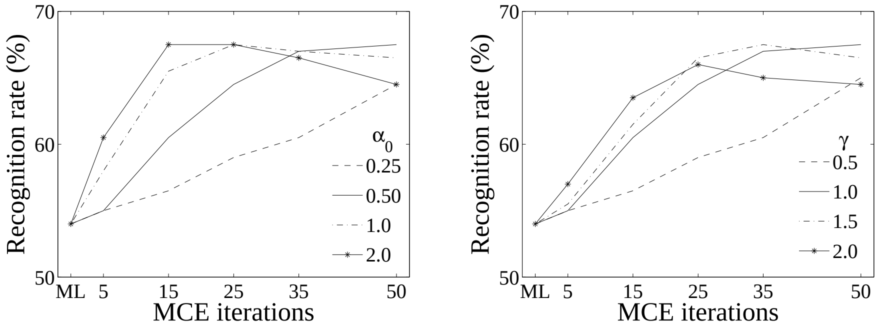


(a)

(b)

Figure 3.7: *Recognition rates for phonemes /eh/ and /ih/: a) using SMF; b) using nSMF. Shown results are the median over ten runs for each tested condition. Error-bars are given by the first and third quartiles of the obtained scores.*

The difference in performance achieved with a different choice of the misclassification function is stressed in the results for phonemes {/eh/,/ih/} shown in Figure 3.7. Scores obtained here with the method based on nSMF are markedly better than those achieved using SMF. For the former the average improvement in the error rate is around 45%, whereas for the latter it is about 20%. A possible explanation of these results relies on the wide dispersion of discriminant function values. As SMF is based just on a difference between these values, it also has a large variability that makes it very difficult to choose a suitable sigmoid to capture many confusable samples to drive the competitive update without picking too much of them. The selected value for γ becomes conservative and then only a small subset of confusable samples are used to trigger the updates, which results in a poorer performance. It must be noticed that this effect is expected to be emphasized as the duration of sequences increases, so that is natural to have better results for the shorter samples from {/b/,/d/}. On the other hand, the misclassification function nSMF introduces a scaling that avoids it to have so much variation in its values, which makes it easier to find a suitable sigmoid to drive the selection of confusable patterns.



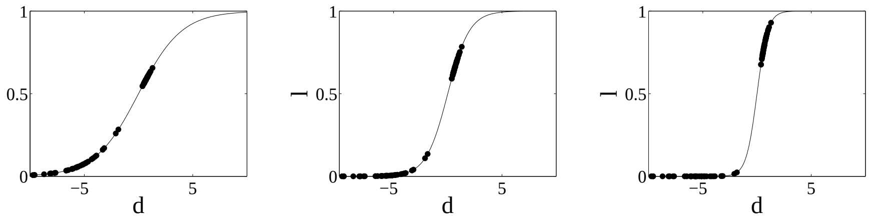
(a) (b)

Figure 3.8: *Sensitivity of recognition rate to changes on the parameters of the MCE/GPD algorithm. a) Varying α_0, with γ fixed. b) Varying γ, with α_0 fixed.*

3.4.3 Sensitivity to parameters of the algorithm

It is interesting to see the effect on the recognition rate when changing the parameters of the MCE/GPD algorithm. Consider the problem of classifying phonemes {/eh/,/ih/}. We first carried out a simple experiment setting $\eta = 4$ and $\gamma = 1$ as in previous tests, and changed α_0 to take values {0.25, 0.50, 1.0, 2.0}. Obtained results are shown in Figure 3.8.a). It can be seen that for this dataset recognition rates attain a bound at 67.5% for all conditions, but they differ in the speed they do it with. The smaller learning rate shows the lowest increase in recognition rate when increasing the number of iterations of the learning algorithm. Increasing α_0 speeds up the process, but it can be seen also that it can lead to overfitting. This situation is common to all gradient-based techniques as the one proposed here. The optimal value of α_0 depends on the data and the size of the training sample. Some rough guidelines to choose this parameter are stated in [69], taking into account the variability of the sample.

A similar effect can be seen in Figure 3.8.b), but varying γ and letting α_0 and η fixed. Nevertheless, the reason is quite different. Parameter γ determines the rate of change of the loss aproximation. For small values of γ, the sigmoid grows slowly from $\ell = 0$ to $\ell = 1$ and much of the training samples result in values of the misclassification function that fall in the raising segment of the sigmoid. In this case, even well classified sequences trigger strong updates. As γ becomes large, the raising segment of the sigmoid gets sharper and less cases fall in this region. Thus,



(a) (b) (c)

Figure 3.9: *Location of the training sequences on the loss function for different values of parameter γ, using nSMF: a) $\gamma = 0.5$, b) $\gamma = 1$; and c) $\gamma = 2$.*

well classified observations introduce a much weaker change on the parameters. At the same time, when nSMF is used as the misclassification function, small values of γ make misclassified cases fall in a narrow segment of the sigmoid, as seen in Figure 3.9. They give rise to updates with similar strength regardless the confusability of the training sequence. As γ becomes larger, misclassified cases occupy a broader region of the sigmoid, triggering updates that depend more on confusability.

3.4.4 Multiclass phoneme recognition

To further assess the proposed discriminative training method for the HMM-HMT model, a new speech recognition task including the whole set of phonemes was carried out. In this experiment, only the MCE approach based on the misclassification function nSMF was taken into account, as consistently better results were found for this choice in the previous task. Ten training sets picked at random were considered and a replicate of the experiment was run for each of them. The testing set remained fixed for all runs. Both the training sets and the testing set were build randomly taking 200 sequences from each class. The same learning rate was used for all the parameters in the models. The initial rate α_0 was chosen to be the largest value that gave a monotonic improvement in recognition rate as a function of the number of iterations of the MCE algorithm, when using a separate set of sequences both for training and testing. This was checked in preliminary runs. During the experiments, this learning rate was linearly decreased from $\alpha_\tau = \alpha_0$ at the first iteration to $\alpha_\tau = 0$ at the end of the training process.

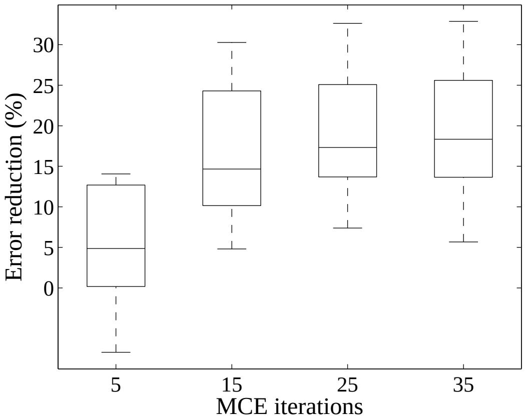


Figure 3.10: *Error rate improvement over standard ML training using the proposed MCE approach to train the classifier for the set of five phonemes. The misclassification function nSMF was used in this experiment. Initial recognition rates using ML estimates are around* 37% *for the considered phoneme set.*

Obtained results are shown in Figure 3.10. A monotonic improvement in the error rate is achieved as more iterations over the whole training set are added to the discriminative training process. After 35 iterations, the average error rate reduction is about 18%. Most of the improvement, however, occurs up to 25 iterations of the MCE algorithm, reducing the error rate around a 17.25% at this level. The variance in the obtained rates remains fairly the same with the increased number of iterations. Analysis of individual runs reveals that for some training sets performance degrades with the first iterations of the algorithm and then starts to improve as more iterations are carried out. Furthermore, three of the ten runs show that the achieved score starts to decrease slowly at 35 iterations, suggesting that overfitting could be taking place after this point.

This difficult classification task show a consistent improvement in recognition rate using the proposed method to discriminatively train the HMM-HMT model.

3.5 Concluding remarks

In this chapter, a new discriminative training method was introduced for hidden Markov models in the wavelet domain. The algorithm is based on the

MCE/GPD approach and it allows for training fully non-tied HMM-HMT models. This observation model and feature space required special considerations. It was shown that standard procedures were numerically unfeasible in this scenario, and alternative choices were needed to simulate the classifier decision when the MCE criterion was derived. Assessment of proposed misclassification functions in a simple phoneme recognition task showed that comparing the order of magnitude of the log-likelihoods for competing models was more appealing to this context than simple comparison of their value. This important modification results in a stronger penalty for misclassified patterns, giving rise to a corrective characteristic that works well in this context. Speech recognition experiments show that the proposed method achieves consistent improvements on recognition rates over training with the standard EM algorithm only.

CHAPTER 4

Discriminative dimension reduction: a sufficiency approach

4.1 Introduction

When parametric models for $\mathbf{X}|Y$ are estimated using maximum likelihood, likelihood-based supervised dimension reduction can be consistently embedded into this learning framework. For GHMM-based classifiers, the techniques most widely used in applications are the subspace projection methods proposed in [57, 56, 83]. They are built upon reduction methods for Gaussian data and pursue likelihood approaches to linear discriminant analysis (LDA) and heteroscedastic linear discriminant analysis (HLDA). But do they retain all the discriminative information that is contained in the original data? If they do, are the obtained subspaces the smallest that show that conservation property?

In this chapter we address these questions under the framework of *sufficient dimension reduction* (SDR), which explicitly accounts for loss of information in the context of a particular task [60, 18]. We show that both LDA and HLDA actually can obtain an optimal subspace projection in the sense of sufficiency for classification, but under some strong constraints on the covariance structure of the class models. In addition, we show that when seen from the sufficiency point of view, HLDA obtains a subspace that may not be minimal. As a remedy, we propose a new linear transformation that satisfies the same covariance constraints HLDA does, but spans the smallest linear subspace that retains all the information about Y. When heteroscedastic data is not constrained to a special covariance structure, we show that there is another estimator derived under sufficiency that provides a more proper way to deal with this type of data and thus it is able to outperform HLDA. The theory and algorithms are developed under the assumption that the dimension d of the retained subspace is known. Nevertheless, theory allows to provide methods for inference on d. We review some of these methods, which also help to ground the selection of d on a principled basis and can serve as alternatives to computationally demanding cross-validation tests.

The chapter is organized as follows. We start by briefly reviewing LDA and HLDA in Section 4.2. In Section 4.3 we review the basics of sufficient dimension reduction, and restate the main results derived for normal models. We then analyze LDA and HLDA from the point of view of sufficiency in Section 4.4. In Section 4.5 we focus on inference methods for the dimension of the retained subspace. We review likelihood ratio tests, information criteria, and permutation tests, which can serve as alternatives to cross-validation estimation of classification errors. Simulations illustrate our points in Section 4.6. Finally, in Section 4.7 we show how these SDR methods originally derived for conditional normal models can be extended to GHMM.

4.2 Existing methods for linear dimension reduction

In this section we briefly review the basics of LDA and HLDA. For convenience, we summarize some notation now. For $\mathbf{A} \in \mathbb{R}^{p\times p}$ and a subspace $\mathcal{S} \subseteq \mathbb{R}^p$, $\mathbf{A}\mathcal{S} \equiv \{\mathbf{A}\mathbf{x} : \mathbf{x} \in \mathcal{S}\}$. $\mathbf{P}_{\mathcal{S}}$ indicates the orthogonal projection onto the subspace $\mathcal{S}$ in the usual inner product, and $\mathbf{Q}_{\mathcal{S}} = \mathbf{I} - \mathbf{P}_{\mathcal{S}}$ is the projection onto its orthogonal complement. In addition, let $\mathbf{V}_d(\mathbf{A})$ stand for the matrix whose columns are the leading d-eigenvectors of the symmetric positive definite matrix $\mathbf{A}$, meaning that they correspond to the d largest eigenvalues of $\mathbf{A}$. Also, assume in the following that we have N_y i.i.d. observations $\{(Y_i = y, \mathbf{X}_i)\}$ for each class $y = 1, 2, \dots, h$, with $N = \sum_y N_y$, let $\boldsymbol{\mu}_y = \mathrm{E}(\mathbf{X}|Y = y)$, $\boldsymbol{\Delta}_y = \mathrm{var}(\mathbf{X}|Y = y)$, $\boldsymbol{\mu} = \mathrm{E}(\boldsymbol{\mu}_Y)$, $\boldsymbol{\Delta} = \mathrm{E}(\boldsymbol{\Delta}_Y)$ and consider statistics $\tilde{\boldsymbol{\mu}}_y = N_y^{-1}\sum_{i=1}^{N_y}\mathbf{X}_i$, $\tilde{\boldsymbol{\Delta}}_y = N_y^{-1}\sum_{i=1}^{N_y}(\mathbf{X}_i -$

$\tilde{\boldsymbol{\mu}}_y)(\mathbf{X}_i - \tilde{\boldsymbol{\mu}}_y)^T$, $\tilde{\boldsymbol{\mu}} = N^{-1}\sum_y N_y \tilde{\boldsymbol{\mu}}_y$, and $\tilde{\boldsymbol{\Delta}} = N^{-1}\sum_y N_y \tilde{\boldsymbol{\Delta}}_y$. Finally, for a parameter γ, let $\hat{\gamma}$ refer to its ML estimator.

4.2.1 Linear discriminant analysis

The best known of supervised dimension reduction methods is Fisher's LDA [40]. It aims to separate classes as far as possible by maximizing the ratio of between-class scatter to average within-class scatter in the transformed space. The transformation matrix $\boldsymbol{\rho}_{\mathrm{LDA}}$ is then determined by maximizing the criterion

$$\mathcal{J}_F(\boldsymbol{\rho}) = \mathrm{tr}\{(\boldsymbol{\rho}^T \tilde{\boldsymbol{\Delta}} \boldsymbol{\rho})^{-1}(\boldsymbol{\rho}^T \mathbf{B} \boldsymbol{\rho})\}, \tag{4.1}$$

where $\mathbf{B} = N^{-1}\sum_{y=1}^{h} N_y(\tilde{\boldsymbol{\mu}}_y - \tilde{\boldsymbol{\mu}})(\tilde{\boldsymbol{\mu}}_y - \tilde{\boldsymbol{\mu}})^T$ is the so-called between-class covariance matrix. Optimization of $\mathcal{J}_F$ boils down to finding the eigenvalue decomposition of $\tilde{\boldsymbol{\Delta}}^{-1/2}\mathbf{B}\tilde{\boldsymbol{\Delta}}^{-1/2}$. Doing this we get

$$\boldsymbol{\rho}_{\mathrm{LDA}} = \tilde{\boldsymbol{\Delta}}^{-1/2}\mathbf{V}_d(\tilde{\boldsymbol{\Delta}}^{-1/2}\mathbf{B}\tilde{\boldsymbol{\Delta}}^{-1/2}). \tag{4.2}$$

As the rank of $\mathbf{B}$ is $h-1$, we can find at most $\min(h-1, p)$ discriminant directions.

While it is not necessary to make restrictive assumptions on $\mathbf{X}|Y$ to derive $\boldsymbol{\rho}_{\mathrm{LDA}}$ in this way, it is well-known that this projection method achieves the best results when $\mathbf{X}|Y$ is normally distributed and all within-class covariance matrices are the same. This observation motivated efforts to understand $\boldsymbol{\rho}_{\mathrm{LDA}}$ as a ML estimator. Such interpretation when $\mathbf{X}|(Y=y) \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Delta})$ is given in [10].

4.2.2 Heteroscedastic linear discriminant analysis

Several extensions to LDA have been proposed to deal with the nonconstant variance case [57, 83, 26, 77, 64, 63]. We are concerned here only with those based on maximum likelihood estimation, so that they can be consistently embedded into HMM training. Probably the best known of these methods is that introduced in [57], which we will simply refer to as HLDA. Their derivation is as follows. Assume $\mathbf{X}|(Y=y) \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Delta}_y)$ and consider a full-rank linear transformation of $\mathbf{X}$ with a matrix $\boldsymbol{\Theta} = (\boldsymbol{\rho}_{\mathrm{HLDA}}, \boldsymbol{\rho}_0)$ so that $\boldsymbol{\Theta}^T\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}_y^*, \boldsymbol{\Delta}_y^*)$, with

$$\boldsymbol{\mu}_y^* = \begin{pmatrix} \boldsymbol{\rho}^T \boldsymbol{\mu}_y \\ \boldsymbol{\rho}_0^T \boldsymbol{\mu} \end{pmatrix} \qquad \boldsymbol{\Delta}_y^* = \begin{pmatrix} \boldsymbol{\Omega}_y & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega}_0 \end{pmatrix}.$$

In this way, $\boldsymbol{\rho}_{\text{HLDA}}^T \mathbf{X}$ is independent of $\boldsymbol{\rho}_0^T \mathbf{X}$ and the latter is constant for all classes y. Thus, $\boldsymbol{\rho}_0^T \mathbf{X}$ does not carry any discriminative information and can be ignored for classification. Without loss of generality, assume $\boldsymbol{\Theta}$ is an orthogonal matrix and that $\boldsymbol{\rho}_{\text{HLDA}}$ is semi-orthogonal. From [57] the optimum matrix $\boldsymbol{\Theta}$ maximizes the log-likelihood function

$$\mathcal{L}_{\text{HLDA}}(\boldsymbol{\Theta}) \;=\; -\frac{N}{2}\log|\boldsymbol{\rho}_0^T \tilde{\boldsymbol{\Sigma}} \boldsymbol{\rho}_0| - \frac{1}{2}\sum_{y=1}^{h} n_y \log|\boldsymbol{\rho}_{\text{HLDA}}^T \tilde{\boldsymbol{\Delta}}_y \boldsymbol{\rho}_{\text{HLDA}}|. \tag{4.3}$$

The optimum does not have a closed-form solution, so numerical techniques must be employed [57, 42]. Notice that in this derivation, beginning with normality for $\mathbf{X}|Y$, restrictions are imposed in the transformed feature space, not in the original space of $\mathbf{X}$. Also, the models assumed in the transformed space are strongly structured to allow statistical independence between $\boldsymbol{\rho}_{\text{HLDA}}^T \mathbf{X}$ and $\boldsymbol{\rho}_0^T \mathbf{X}$.

It is also interesting to analyze the case in which $\boldsymbol{\Omega}_y = \boldsymbol{\Omega}$; that is, when it is the same for all classes. Now it is obvious that $\boldsymbol{\Delta}_y = \boldsymbol{\Delta}$ for all y; then no part of the covariance matrices has any discriminative information. The log-likelihood function (4.3) reduces to

$$\mathcal{L}(\boldsymbol{\Theta}) \;=\; -\frac{N}{2}\log|\boldsymbol{\rho}_0^T \tilde{\boldsymbol{\Sigma}} \boldsymbol{\rho}_0| - \frac{N}{2}\log|\boldsymbol{\rho}^T \tilde{\boldsymbol{\Delta}} \boldsymbol{\rho}|. \tag{4.4}$$

It is stated in [96, 57] that maximization of this function gives rise to $\boldsymbol{\rho}_{\text{LDA}}$, allowing us to interpret it as a special case of $\boldsymbol{\rho}_{\text{HLDA}}$ when all covariance matrices are the same. We think this statement is wrong. For all y, $\boldsymbol{\Delta}_y^* = \boldsymbol{\Delta}^*$ will still have a block-diagonal structure

$$\boldsymbol{\Delta}^* = \begin{pmatrix} \boldsymbol{\Omega} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega}_0 \end{pmatrix}.$$

Thus, even in this case $\boldsymbol{\Delta}^*$ induces a particular structure for the covariance matrix $\boldsymbol{\Delta}$, not just being the same for all classes. That is, when $\boldsymbol{\Delta}_y = \boldsymbol{\Delta}$ for all classes but $\boldsymbol{\Delta}$ is an arbitrary covariance matrix without this structure, we cannot assure $\boldsymbol{\rho}_{\text{LDA}} = \boldsymbol{\rho}_{\text{HLDA}}$.

In fact, it can be shown that (4.4) is induced by special assumptions on the normal class models. The corresponding model is known as *extended principal fitted components* in the Statistics literature [18]. Furthermore, it is stated there that there is not an analytical solution to (4.4) and numerical optimization has to be used [18].

4.3 Sufficient dimension reduction

Sufficient dimension reduction is a methodology that deals explicitly with information retention. In this section we review the basics of the sufficiency framework and restate the main results derived for normal models.

4.3.1 Basics

For a response variable $Y \in \mathbb{R}$ and a set of features or predictors $\mathbf{X} \in \mathbb{R}^p$, the following definition formalizes the notion of a sufficient dimension reduction [18]:

Definition: *A reduction $R : \mathbb{R}^p \to \mathbb{R}^d$, with $d \leq p$ is sufficient if it satisfies one of the following conditions:*

(i) $Y|\mathbf{X} \sim Y|R(\mathbf{X})$

(ii) $\mathbf{X}|(Y, R(\mathbf{X})) \sim \mathbf{X}|R(\mathbf{X})$

(iii) $(\mathbf{X} \perp Y) \mid R(\mathbf{X})$

Notice that each of these conditions conveys the idea that $R(\mathbf{X})$ carries all the information about Y that is contained in $\mathbf{X}$. One may be more useful than the others depending on the stochastic nature of Y and $\mathbf{X}$, but they are equivalent when $(Y, \mathbf{X})$ has a joint distribution, as is usually assumed with Bayes classifiers.

In this work we deal only with linear reductions of the form $R(\mathbf{X}) = \boldsymbol{\rho}^T\mathbf{X}$. Note that the full feature vector $\mathbf{X}$ is always a sufficient reduction. Thus, the essential tasks in SDR are to characterize and estimate the *smallest* sufficient reduction. In addition, if $\boldsymbol{\rho}^T\mathbf{X}$ is a sufficient reduction and $\boldsymbol{\eta} \in \mathbb{R}^{d\times d}$ is a non-singular matrix, then $\boldsymbol{\eta}\boldsymbol{\rho}^T\mathbf{X}$ is also a sufficient reduction. Thus, $\boldsymbol{\rho}$ is not unique and what really makes sense to identify is the subspace spanned by the columns of $\boldsymbol{\rho}$. This subspace $\mathcal{S}_{\boldsymbol{\rho}} = \text{span}(\boldsymbol{\rho})$ is called a *sufficient dimension reduction subspace*. Under mild but non-neglible conditions, the intersection of all sufficient dimension reduction subspaces is also a sufficient dimension reduction subspace and thus it is the smallest one. It is called the *central subspace* [16, 17] and it is the inferential target in SDR. From now on, unless stated otherwise, $\boldsymbol{\rho}$ will be a basis matrix for the central subspace.

Here we are interested in the case where $\mathbf{X}|Y$ is normally distributed with parameters $\boldsymbol{\mu}_y$ and $\boldsymbol{\Delta}_y$. Under this model, the central subspace exists and we can employ a likelihood function to estimate it from the data. Then, maximum likelihood estimation guarantees $\sqrt{N}$ consistency and also asymptotical efficiency when the likelihood accurately describes the data.

It might be argued, however, that the definition stated above for sufficient dimension reduction is not focussed explicitly in classification. In a classification framework, we are interested actually in finding a classification rule to assign a label $Y = y$ to each feature vector $\mathbf{X}$. If $f(\mathbf{X}) : \mathbb{R}^p \to \{1, 2, \ldots, h\}$ is the decision rule, we can think of a reduction as sufficient if given $\mathbf{X} = \mathbf{x}$, $f(\boldsymbol{\rho}^T\mathbf{x}) = f(\mathbf{x})$ for each $\mathbf{x}$ in the feature space. The subspace spanned by the columns of $\boldsymbol{\rho}$ would be then a *central discriminant subspace*[1] [23, 94]. This subspace may be a subset of the central subspace, as we may need less information to discriminate between classes than to describe them accurately. Nevertheless, when using the common Bayes classification rule, it was shown in [23] that this discriminant subspace is identical to the central subspace when class models are Gaussian distributions. Thus, for normally distributed data we can exploit theory recently developed for regression tasks to get further insight into dimension reduction aimed to classification tasks.

4.3.2 Sufficient reductions for normal models

The theory of sufficient dimension reduction for normally distributed data with constant covariance matrix was presented in [18] and further developed in [21]. The extension to general cases with unconstrained covariance was introduced in

[1] In [94] this subspace is referred to as *intrinsic Bayes discriminant subspace*. We prefer the terminology used here to keep it closer to the central subspace widely known in regressions.

[20]. The following theorem, demonstrated in [20], gives necessary and sufficient conditions for a subspace $\mathcal{S}$ to be a dimension reduction subspace.

Theorem 1: Assume that $\mathbf{X}|(Y = y) \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Delta}_y), y = 1, 2, \ldots, h$. Then $\mathcal{S}_{\boldsymbol{\rho}} = \text{span}(\boldsymbol{\rho}) \subset \mathbb{R}^p$ is a sufficient dimension reduction subspace if and only if:

a) $\text{span}(\boldsymbol{\mu}_y - \boldsymbol{\mu}) = \boldsymbol{\Delta}\ \text{span}(\boldsymbol{\rho})$.

b) $Q_{\mathcal{S}_{\boldsymbol{\rho}}}\boldsymbol{\Delta}_y^{-1}$ *does not depend on the class* Y.

This theorem implies that the subspace spanned by $\boldsymbol{\Delta\rho}$ must be an invariant subspace for the deviations $\boldsymbol{\Delta}_y - \boldsymbol{\Delta}$, and that the translated means $\boldsymbol{\mu}_y - \boldsymbol{\mu}$ must fall also in that subspace[2] [20]. Under these conditions, the means and covariance matrices of the class models are

$$\begin{aligned} \boldsymbol{\mu}_y &= \boldsymbol{\mu} + \boldsymbol{\Delta\rho\nu}_y, \\ \boldsymbol{\Delta}_y &= \boldsymbol{\Delta} + \boldsymbol{\Delta\rho}\mathbf{T}_y\boldsymbol{\rho}^T\boldsymbol{\Delta}, \end{aligned} \tag{4.5}$$

for some $\boldsymbol{\nu}_y \in \mathbb{R}^d$ and $\bar{\boldsymbol{\nu}} = \sum_y \boldsymbol{\nu}_y = 0$, $\mathbf{T}_y \in \mathbb{R}^{d\times d}$ and $\sum_y \mathbf{T}_y = \mathbf{0}$, and $d = \dim(\mathcal{S}_{\boldsymbol{\rho}})$. It is important to emphasize that (4.5) are necessary and sufficient conditions derived from Theorem 1 to assure the existence of a linear SDR when $\mathbf{X}|Y \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Delta}_y)$; they are not assumptions set *a priori* to derive the subspace projection method.

Despite this theorem being a main result, in practice we are interested in an estimator for $\mathcal{S}_{\boldsymbol{\rho}}$. Going in that direction, let $\boldsymbol{\rho}$ be a semi-orthogonal basis matrix for $\mathcal{S}_{\boldsymbol{\rho}} \subseteq \mathbb{R}^p$ and let $(\boldsymbol{\rho}, \boldsymbol{\rho}_0) \in \mathbb{R}^{p\times p}$ be an orthogonal matrix. It is shown in [20] that $\mathcal{S}_{\boldsymbol{\rho}}$ is a sufficient dimension reduction subspace if and only if the following two conditions are satisfied for some vectors $\boldsymbol{\nu}_y$

1. $\boldsymbol{\rho}^T\mathbf{X}|(Y = y) \sim \mathcal{N}(\boldsymbol{\rho}^T(\boldsymbol{\mu} + \boldsymbol{\Delta\rho\nu}_y), \boldsymbol{\rho}^T\boldsymbol{\Delta}_y\boldsymbol{\rho})$
2. $\boldsymbol{\rho}_0^T\mathbf{X}|(\boldsymbol{\rho}^T\mathbf{X}, Y = y) \sim \mathcal{N}(\boldsymbol{\rho}_0^T\boldsymbol{\mu} + \mathbf{H}\boldsymbol{\rho}^T(\mathbf{X} - \boldsymbol{\mu}), \mathbf{D})$, with $\mathbf{D} = (\boldsymbol{\rho}_0^T\boldsymbol{\Delta}^{-1}\boldsymbol{\rho}_0)^{-1}$ and $\mathbf{H} = (\boldsymbol{\rho}_0^T\boldsymbol{\Delta\rho})(\boldsymbol{\rho}^T\boldsymbol{\Delta\rho})^{-1}$.

It is clear now that if $\mathcal{S}_{\boldsymbol{\rho}}$ is a dimension reduction subspace, the distribution of $\boldsymbol{\rho}^T\mathbf{X}|(Y = y)$ can depend on Y, but the distribution of $\boldsymbol{\rho}_0^T\mathbf{X}|(\boldsymbol{\rho}^T\mathbf{X}, Y = y)$

[2] $\mathcal{S} \subset \mathbb{R}^p$ is an invariant subspace of $\mathbf{A} \in \mathbb{R}^{p\times p}$ if $\mathbf{A}\mathcal{S} \subseteq \mathcal{S}$.

cannot. Thus, $\boldsymbol{\rho}^T\mathbf{X}$ carries all the information that $\mathbf{X}$ contains about Y and $\boldsymbol{\rho}_0^T\mathbf{X}|\boldsymbol{\rho}^T\mathbf{X}$ does not retain any information about the class and it is irrelevant for classification.

4.3.3 The optimal estimator under sufficiency

With the ingredients stated in the last subsection, we are ready to obtain the MLE of $\boldsymbol{\rho}$. Assume that $\boldsymbol{\rho} \in \mathbb{R}^{p\times d}$ is a semi-orthogonal basis matrix for the smallest dimension reduction subspace. For normally distributed data with means and covariance matrices as in (4.5), the MLE $\boldsymbol{\rho}_{\text{LAD}}$ maximizes the log likelihood function [20]

$$\mathcal{L}_{\text{LAD}}(\boldsymbol{\rho}) = \text{const} + \frac{N}{2}\log|\boldsymbol{\rho}^T\tilde{\boldsymbol{\Sigma}}\boldsymbol{\rho}| - \frac{1}{2}\sum_y N_y \log|\boldsymbol{\rho}^T\tilde{\boldsymbol{\Delta}}_y\boldsymbol{\rho}|. \tag{4.6}$$

This estimator is simply known as *likelihood acquired directions* (LAD). There is not an analytic solution to this maximization problem, so we must employ numerical optimization to find $\boldsymbol{\rho}$ that maximizes $\mathcal{L}_{\text{LAD}}(\boldsymbol{\rho})$. In addition, to guarantee achieving the MLE, all the columns of $\boldsymbol{\rho}$ should be estimated jointly. We recall that the stated result restricts itself to semiorthogonal matrices $\boldsymbol{\rho}$. It is easy to see that for any nonsingular matrix $\mathbf{O} \in \mathbb{R}^{d\times d}$, $\mathcal{L}_{\text{LAD}}(\boldsymbol{\rho}) = \mathcal{L}_{\text{LAD}}(\boldsymbol{\rho}\mathbf{O})$. Thus, the natural parameter space for $\boldsymbol{\rho}$ is the Grassmann manifold of dimension d in $\mathbb{R}^p$ [12].

The LAD estimator is equivariant under full-rank transformation of the features $\mathbf{X}$. That is, if we rescale the observed $\mathbf{X}$ as $\boldsymbol{\eta}^T\mathbf{X}$ prior to estimation, the obtained estimator will be a semi-orthogonal basis matrix for span$(\boldsymbol{\eta}\boldsymbol{\rho})$ provided $\boldsymbol{\eta}$ is a nonsingular matrix. This invariance property does not hold for HLDA, as shown later in Section 4.4.2. In addition, LAD is found to perform well even when the data deviate from normality [20]. In particular, it can be shown that if $\mathrm{E}(\mathbf{X}|\boldsymbol{\rho}^T\mathbf{X})$ is linear and $\mathrm{var}(\mathbf{X}|\boldsymbol{\rho}^T\mathbf{X})$ is a nonrandom matrix, then the subspace spanned by $\tilde{\boldsymbol{\rho}}$ as found by maximizing (4.6) is a consistent estimate of the minimal reduction subspace [20].

4.4 Understanding existing methods under SDR

In this section we wonder if the frequently used methods LDA and HLDA for likelihood-based subspace projection of Gaussian data can be understood under the sufficiency approach, that is, if they do not loose any class-information that was present in the original features. Under what assumptions on the class models do these methods provide sufficient dimension reduction in the sense discussed here? We work on this question in the following paragraphs.

4.4.1 LDA from the sufficiency approach

When $\boldsymbol{\Delta}_y = \boldsymbol{\Delta}$ for all y, condition b) in Theorem 1 becomes trivial, and $\boldsymbol{\rho}^T\mathbf{X}$ is a minimal sufficient reduction if and only if $\text{span}(\boldsymbol{\mu}_y - \boldsymbol{\mu}) = \boldsymbol{\Delta}\,\text{span}(\boldsymbol{\rho})$, with class models being normal distributions with mean $\boldsymbol{\mu}_y = \boldsymbol{\mu} + \boldsymbol{\Delta}\boldsymbol{\rho}\boldsymbol{\nu}_y$ and covariance matrix $\boldsymbol{\Delta}$ for all y [18, 21].

A basis matrix for this minimal dimension reduction subspace can be found by modeling $\boldsymbol{\nu}_y$ [21]. Assume for a moment that Y is a general response variable in $\mathbb{R}$ and let $\mathbf{Y} \in \mathbb{R}^r$ be a vector valued function of Y. Let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{N\times p}$ stand for the whole sample of feature vectors, where each row is an observation, and let $\boldsymbol{\mathcal{X}}_c$ be its centered counterpart. Taking $\boldsymbol{\nu}_y = \boldsymbol{\beta}\mathbf{Y}$, with $\boldsymbol{\beta} \in \mathbb{R}^{d\times(h-1)}$, the centered fitted values $\tilde{\boldsymbol{\mathcal{X}}}$ of the linear multivariate regression of $\boldsymbol{\mathcal{X}}_c|Y$ on $\mathbf{Y}$ have covariance matrix $\tilde{\boldsymbol{\Sigma}}_{fit} = \tilde{\boldsymbol{\mathcal{X}}}^T\tilde{\boldsymbol{\mathcal{X}}}/N$. Define $\tilde{\boldsymbol{\Sigma}}_{res}$ so that $\tilde{\boldsymbol{\Sigma}} = \tilde{\boldsymbol{\Sigma}}_{fit} + \tilde{\boldsymbol{\Sigma}}_{res}$. It is shown in [21] that $\boldsymbol{\rho} = \tilde{\boldsymbol{\Sigma}}_{res}^{-1/2}\mathbf{V}_d(\tilde{\boldsymbol{\Sigma}}_{res}^{-1/2}\tilde{\boldsymbol{\Sigma}}_{fit}\tilde{\boldsymbol{\Sigma}}_{res}^{-1/2})$, with $d \leq \min(h-1, p)$, spans the smallest dimension reduction subspace when $\mathbf{X}|(Y = y)$ is normally distributed with mean $\boldsymbol{\mu}_y = \boldsymbol{\mu} + \boldsymbol{\Delta}\boldsymbol{\rho}\boldsymbol{\beta}\mathbf{Y}$ and covariance matrix $\boldsymbol{\Delta}_y = \boldsymbol{\Delta}$ for all classes y. This reduction is called *principal fitted components* (PFC).

While this development seems more tailored to dimension reduction in regression, we want to emphasize here that it is equally suitable to discrimination tasks. Indeed, when Y represents class labels, the estimator $\boldsymbol{\rho}_{\text{PFC}}$ found in this way resembles $\boldsymbol{\rho}_{\text{LDA}}$. To see this, let $\mathbf{Y} \in \mathbb{R}^{h-1}$ be an indicator multivariate response whose columns designate the class from where the feature vector $\mathbf{X}$ comes. In particular, if $\mathbf{X} = \mathbf{x}$ comes from class $y = k$, the i-th coordinate of $\mathbf{Y}$ takes the value $1 - N_k/N$ if $i = k$ and $-N_i/N$ otherwise. Note that $E\{\mathbf{Y}\} = \mathbf{0}$ with this choice.

With this setting, assume as before that $\mathbf{X}|(Y = y) \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Delta})$ with $\boldsymbol{\mu}_y = \boldsymbol{\mu} + \boldsymbol{\Delta}\boldsymbol{\rho}\boldsymbol{\beta}\mathbf{Y}$ so that a sufficient reduction exists. The sample covariance matrix of the fitted values $\tilde{\boldsymbol{\Sigma}}_{fit} = \tilde{\boldsymbol{\mathcal{X}}}^T\tilde{\boldsymbol{\mathcal{X}}}/n$ is the sample between-class scatter matrix $\mathbf{B}$ defined above. As the marginal sample covariance matrix is $\tilde{\boldsymbol{\Sigma}} = \mathbf{B} + \boldsymbol{\Delta} = \tilde{\boldsymbol{\Sigma}}_{fit} + \tilde{\boldsymbol{\Sigma}}_{res}$, $\tilde{\boldsymbol{\Sigma}}_{res}$ takes the place of $\widehat{\boldsymbol{\Delta}}$. Then $\widehat{\boldsymbol{\rho}} = \widehat{\boldsymbol{\Delta}}^{-1/2}\mathbf{V}_d(\widehat{\boldsymbol{\Delta}}^{-1/2}\mathbf{B}\widehat{\boldsymbol{\Delta}}^{-1/2})$, with $d \leq \min(h-1, p)$, is a basis matrix for the smallest dimension reduction subspace. The relationship with $\boldsymbol{\rho}_{\text{LDA}}$ in (4.2) is clear.

Note this result provides both a maximum likelihood derivation of LDA and a sufficiency interpretation for it. Although there exists other developments to cast the LDA projection in a likelihood framework [10], the one presented here gives sufficient conditions on the distribution of $\mathbf{X}|(Y = y)$ so that $\boldsymbol{\rho}_{\text{LDA}}^T\mathbf{X}$ retains all the information about Y that is contained in $\mathbf{X}$. As a consequence, this interpretation allows us to choose a dimension $d \leq \min(h-1, p)$ for the minimal dimension reduction subspace using tools derived from theory.

4.4.2 HLDA from the sufficiency point of view

We saw in Section 2.2 that HLDA was derived in [57] assuming a particular model for the transformed features $\boldsymbol{\Theta}^T\mathbf{X}$. To gain insight into this method under the sufficiency approach, we need the model induced by these assumptions back in the original space of the features $\mathbf{X}$.

Let $\boldsymbol{\Theta} = (\boldsymbol{\rho}, \boldsymbol{\rho}_0)$ be an orthogonal matrix with $\boldsymbol{\rho} \in \mathbb{R}^{p \times d}$. It is easy to see that we get the HLDA assumptions $\boldsymbol{\rho}^T\mathbf{X}|Y = y \sim \mathcal{N}(\boldsymbol{\rho}^T\boldsymbol{\mu}_y, \boldsymbol{\Omega}_y)$ and $\boldsymbol{\rho}_0^T\mathbf{X}|(\boldsymbol{\rho}^T\mathbf{X}, Y = y) \sim \mathcal{N}(\boldsymbol{\rho}_0^T\boldsymbol{\mu}, \boldsymbol{\Omega}_0)$ if and only if $\mathbf{X}|(Y = y)$ is normally distributed with mean and covariance matrix

$$\begin{aligned} \boldsymbol{\mu}_y &= \boldsymbol{\mu} + \boldsymbol{\rho}\boldsymbol{\nu}_y \,, \\ \boldsymbol{\Delta}_y &= \boldsymbol{\rho}\boldsymbol{\Omega}_y\boldsymbol{\rho}^T + \boldsymbol{\rho}_0\boldsymbol{\Omega}_0\boldsymbol{\rho}_0^T . \end{aligned} \tag{4.7}$$

In addition, it is clear that $\boldsymbol{\Delta} = \boldsymbol{\rho}\boldsymbol{\Omega}\boldsymbol{\rho}^T + \boldsymbol{\rho}_0\boldsymbol{\Omega}_0\boldsymbol{\rho}_0^T$, where $\boldsymbol{\Omega} = \sum_y \boldsymbol{\Omega}_y$. This structure implies that the subspace spanned by $\boldsymbol{\rho}$ reduces $\boldsymbol{\Delta}$, i.e. there exists a matrix $\mathbf{C} \in \mathbb{R}^{d \times d}$ so that $\boldsymbol{\Delta}\boldsymbol{\rho} = \boldsymbol{\rho}\mathbf{C}$. Then, rewritting $\boldsymbol{\nu}_y = \mathbf{C}\boldsymbol{\gamma}_y$ and $\boldsymbol{\Omega}_y - \boldsymbol{\Omega} = \mathbf{C}\mathbf{T}_y\mathbf{C}^T$, we get $\boldsymbol{\mu}_y = \boldsymbol{\mu} + \boldsymbol{\Delta}\boldsymbol{\rho}\boldsymbol{\gamma}_y$ and $\boldsymbol{\Delta}_y = \boldsymbol{\Delta} + \boldsymbol{\Delta}\boldsymbol{\rho}\mathbf{T}_y\boldsymbol{\rho}^T\boldsymbol{\Delta}$. We see that $\boldsymbol{\rho} = \boldsymbol{\rho}_{\text{HLDA}}$ satisfies (4.5) and as a result it is a special case of LAD and then it is a basis matrix for a dimension reduction subspace. Thus, HLDA estimates a sufficient reduction provided $\mathbf{X}|(Y = y)$ is normally distributed with mean $\boldsymbol{\mu}_y = \boldsymbol{\mu} + \boldsymbol{\rho}\boldsymbol{\nu}_y$ and covariance matrix $\boldsymbol{\Delta}_y = \boldsymbol{\rho}\boldsymbol{\Omega}_y\boldsymbol{\rho}^T + \boldsymbol{\rho}_0\boldsymbol{\Omega}_0\boldsymbol{\rho}_0^T$.

The derivation above emphasizes that HLDA as introduced in [57] can be regarded as an extension of LDA for heteroscedastic data *with constrained covariance matrix*. As a consequence, it does not seem suitable to consider HLDA as a general extension of Fisher's LDA for every type of heteroscedastic data. On the other hand, the LAD model discussed in Section 4.3.3 provides that natural extension allowing for class models with unconstrained covariance matrices. In addition, the strong independence assumed in the transformed domain between $\boldsymbol{\rho}_{\text{HLDA}}^T\mathbf{X}$ and $\boldsymbol{\rho}_0^T\mathbf{X}$ will no longer hold, in general, after rescaling the features with an arbitrary nonsingular matrix $\boldsymbol{\eta}$. Thus, unlike the LAD estimator, the HLDA estimator is not equivariant under full rank transformation of the features. This is an important point that becomes clear with the simulations in Section 4.6.

4.4.3 The minimality question

We saw in Section 4.4.2 that HLDA can give a sufficient linear reduction provided the data has a particular covariance structure. Nevertheless, it is interesting to recall that if a dimension reduction subspace is a subset of a bigger subspace, then the larger subspace is also a dimension reduction subspace. Thus, there exist sufficient dimension reductions that are nonminimal; that is, we could expect to reduce the retained subspace even further. So we turn now to the question of minimality of reductions obtained using HLDA: are the retained directions the fewest linear combinations of the features that retain all the information about the class or can we find a smaller linear subspace that still conserves all of that information?

The answer seems rather evident at this point. From our previous discussions, it is easy to see that in general we cannot expect the subspace spanned by $\boldsymbol{\rho}_{\text{HLDA}}$ to be the smallest dimension reduction subspace, although it will be so when the required covariance structure holds. We focus on giving an intuitive explanation here. The general lack of minimality of the HLDA estimator is due to the particular covariance structure of the assumed class models. The transformation needs to accomodate all the class-specific information there is in $\boldsymbol{\Delta}_y$ into matrices $\boldsymbol{\Omega}_y$, *and* achieve statistical independence between $\boldsymbol{\rho}_{\text{HLDA}}^T\mathbf{X}$ and $\boldsymbol{\rho}_0^T\mathbf{X}$. This fact determines the dimension of span$(\boldsymbol{\rho}_{\text{HLDA}})$, as $\boldsymbol{\rho}_{\text{HLDA}}$ must capture this structure in the covariance. This is always possible with $d = p$, albeit the reduction is no longer useful.

As the dimension of $\boldsymbol{\Omega}_y$ grows, it is more probable that the smallest dimension reduction subspace is a subset of span$(\boldsymbol{\rho}_{\text{HLDA}})$. Assume that the dimension of span$(\boldsymbol{\rho}_{\text{HLDA}})$ is actually u, and that $\boldsymbol{\alpha} \in \mathbb{R}^{p\times d}$, $d \leq u \leq p$, is a semiorthogonal basis matrix for the smallest sufficient dimension reduction subspace (we can infer about both u and d as we will see in Section 4.5). If span$(\boldsymbol{\alpha}) \subseteq$ span$(\boldsymbol{\rho}_{\text{HLDA}})$, then there exist a semi-orthogonal matrix $\mathbf{A} \in \mathbb{R}^{u\times d}$ so that $\boldsymbol{\alpha} = \boldsymbol{\rho}_{\text{HLDA}}\mathbf{A}$. Thus, HLDA provides a minimal sufficient dimension reduction only when $u = d$. If this is not the case, HLDA will still be able to achieve a sufficient dimension reduction $\boldsymbol{\rho}_{\text{HLDA}}^T\mathbf{X} \in \mathbb{R}^u$, but it will not be minimal. On the other hand, LAD always estimates the smallest linear reduction, so that span$(\boldsymbol{\rho}_{\text{LAD}}) \subseteq$ span$(\boldsymbol{\rho}_{\text{HLDA}})$. In practice, the effect of this is that HLDA often needs to retain more directions than LAD to properly account for all the discriminative information.

4.4.4 A new estimator LAD2

Assuming the HLDA model (4.7) and recalling that $\boldsymbol{\rho} \in \mathbb{R}^{p\times u}$ is a sufficient reduction, it follows that $\boldsymbol{\Delta}$ has a structure $\boldsymbol{\Delta} = \boldsymbol{\rho}\boldsymbol{\Omega}\boldsymbol{\rho}^T + \boldsymbol{\rho}_0\boldsymbol{\Omega}_0\boldsymbol{\rho}_0^T$, where $\boldsymbol{\Omega} = \mathrm{E}(\boldsymbol{\Omega}_Y)$. If the minimal reduction, that is the central subspace, is span$(\boldsymbol{\alpha})$, then $\boldsymbol{\alpha} = \boldsymbol{\rho}\mathbf{A}$ for some semiorthogonal $\mathbf{A} \in \mathbb{R}^{u\times d}$, with $d \leq u$. Using this statement and (4.5) we get

$$\begin{aligned}
\boldsymbol{\mu}_y &= \boldsymbol{\mu} + \boldsymbol{\Delta}\boldsymbol{\alpha}\boldsymbol{\nu}_y \\
&= \boldsymbol{\mu} + \boldsymbol{\rho}\boldsymbol{\Omega}\mathbf{A}\boldsymbol{\nu}_y, \\
\boldsymbol{\Delta}_y &= \boldsymbol{\Delta} + \boldsymbol{\Delta}\boldsymbol{\alpha}\mathbf{T}_y\boldsymbol{\alpha}^T\boldsymbol{\Delta} \\
&= \boldsymbol{\rho}\boldsymbol{\Omega}\boldsymbol{\rho}^T + \boldsymbol{\rho}_0\boldsymbol{\Omega}_0\boldsymbol{\rho}_0^T + \boldsymbol{\rho}\boldsymbol{\Omega}\mathbf{A}\mathbf{T}_y\mathbf{A}^T\boldsymbol{\Omega}\boldsymbol{\rho}^T. \qquad (4.8)
\end{aligned}$$

From the previous discussion, the semi-orthogonal basis matrix $\boldsymbol{\alpha}$ can be regarded as a special case of $\boldsymbol{\rho}_{\text{LAD}}$. Nevertheless, the LAD reduction does not recognize the special structure of the covariance matrices. If model (4.8) actually holds for the data, we can look for a more efficient reduction by taking the covariance constraints into account. To do so *and* achieve a minimal sufficient reduction, we need to estimate $\boldsymbol{\rho}$ and $\mathbf{A}$ jointly by maximizing the likelihood function

$$\begin{aligned}\mathcal{L}_{\text{LAD2}}(\boldsymbol{\rho}, \mathbf{A}) \;&=\; \text{const} - \frac{N}{2}\log|\boldsymbol{\rho}^T\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{\rho}| - \frac{N}{2}\log|\boldsymbol{\rho}^T\tilde{\boldsymbol{\Sigma}}\boldsymbol{\rho}| + \\ &+\frac{N}{2}\log|\mathbf{A}^T\boldsymbol{\rho}^T\tilde{\boldsymbol{\Sigma}}\boldsymbol{\rho}\mathbf{A}| - \frac{1}{2}\sum_y N_y \log|\mathbf{A}^T\boldsymbol{\rho}^T\tilde{\boldsymbol{\Delta}}_y\boldsymbol{\rho}\mathbf{A}|,\end{aligned} \tag{4.9}$$

with $\boldsymbol{\rho}$ in the Grassmann manifold of dimension u in $\mathbb{R}^p$, and $\mathbf{A}$ in the Grassmann manifold of dimension d in $\mathbb{R}^u$. The proof is left to the Appendix B. We will refer to this estimator as LAD2 and will denote it by $\boldsymbol{\rho}_{\text{LAD2}}$.

A priori, when the data is normally distributed with this structure, estimating $\boldsymbol{\rho}$ and $\mathbf{A}$ in this way should be more efficient than using LAD, since when $u < p$ there are less degrees of freedom in these computations than in LAD. It is interesting to recall that if we knew $\boldsymbol{\rho}$, $\mathbf{A}$ would reduce to the LAD estimator for the transformed features $\boldsymbol{\rho}^T\mathbf{X}$. As $\boldsymbol{\rho}$ provides the same covariance structure as $\boldsymbol{\rho}_{\text{HLDA}}$, we can approximate the solution applying HLDA first to the features $\mathbf{X}$ and then obtaining the LAD estimator $\mathbf{A}_{\text{LAD}}$ for the transformed data $\boldsymbol{\rho}^T\mathbf{X}|Y$. In this way, $\boldsymbol{\rho}_{\text{HLDA}}\mathbf{A}_{\text{LAD}}$ can serve as an estimator of $\boldsymbol{\rho}_{\text{LAD2}}$, though not being the MLE. In addition, note that when $u = d$, $\mathbf{A}$ is an orthogonal matrix and maximizing (4.9) over $\boldsymbol{\rho}$ gives $\boldsymbol{\rho}_{\text{HLDA}}$ again.

4.4.5 Connections to other methods for heteroscedastic data

While we have focused our attention on HLDA due to its historical importance in applications, in particular for speech technologies, there are other related methods that deserve consideration. In [83], a projection for heteroscedastic data is proposed by generalizing Fisher's criterion as

$$\mathcal{J}_{\text{HDA}} = \prod_{y=1}^{h}\left(\frac{|\boldsymbol{\rho}\mathbf{B}\boldsymbol{\rho}^T|}{|\boldsymbol{\rho}\tilde{\boldsymbol{\Delta}}_y\boldsymbol{\rho}^T|}\right)^{N_y}. \tag{4.10}$$

Taking the log and rearranging terms, maximizing $\mathcal{J}_{\text{HDA}}$ amounts to maximizing ([83] eq. 3)

$$H(\boldsymbol{\rho}) = -\sum_{y=1}^{h} N_y \log|\boldsymbol{\rho}\tilde{\boldsymbol{\Delta}}_y\boldsymbol{\rho}^T| + N\log|\boldsymbol{\rho}\mathbf{B}\boldsymbol{\rho}^T|. \tag{4.11}$$

As $H(\boldsymbol{\rho})$ differs from (4.6) just on a term that does not depend on the transformation, it is clear that optimization of this objective function gives the same estimator as LAD. Nevertheless, one is derived through an heuristic while the other is driven explicitly by information retention as a goal.

The dimension reduction method proposed in [97] is also related to LAD under some special conditions. It aims at extending Fisher's LDA to nonparameteric densities by sequentially maximizing a generalized log-likelihood ratio statistic in a fixed direction $\boldsymbol{\alpha}$. For normal class models, this criterion reduces to [98]

$$\mathrm{LR}(\boldsymbol{\alpha}) = \sum_{y=1}^{h} \frac{N_y}{N} (\log \boldsymbol{\alpha}^T \boldsymbol{\Sigma} \boldsymbol{\alpha} - \log \boldsymbol{\alpha}^T \boldsymbol{\Delta}_y \boldsymbol{\alpha}). \tag{4.12}$$

After the first unit vector is obtained, say $\boldsymbol{\alpha}_1$, the method proceeds by maximizing the same objective function with the added constraint $\boldsymbol{\alpha}_2^T \boldsymbol{\alpha}_1 = 0$, and so on. It is easy to see that $\boldsymbol{\alpha}_1$ is identical to $\boldsymbol{\rho}_{\mathrm{LAD}}$ when the dimension of the central subspace is assumed to be $d = 1$. Nevertheless, adding a second dimension $\boldsymbol{\alpha}_2$ in this way, the subspace spanned by the matrix $(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$ is not equivalent to $\mathrm{span}(\boldsymbol{\rho}_{\mathrm{LAD}})$ for $d = 2$, with both columns of $\boldsymbol{\rho}_{\mathrm{LAD}}$ estimated jointly. An example with real data is used in [19] to illustrate that while $\mathrm{span}(\boldsymbol{\rho}_{\mathrm{LAD}})$ can capture all the structure and separate well the classes with just two directions, $(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$ cannot perform comparably and lead to overlapped clusters of projected features. The central point we want to stress is that the performance of a given dimension reduction method depends on both the objective function being optimized and the procedure used to carry it out. In particular, sequential optimization may lead to different estimates than joint optimization of the likelihood. The MLE of (4.6) is guaranteed using joint maximization but not proceeding sequentially. The same is true for the methods for infering about the dimension d of the central subspace we review in the following section.

4.5 Choosing the dimension of the reduction

In previous sections we assumed that we knew the dimension d of the smallest linear subspace that retained all the class information. In practice, we do not know this quantity and we have to infer it from the data. Most dimension reduction methods rely on an exhaustive approach to infer the dimension of the retained subspace. In them, a sequence of reductions of increasing size are tested based on some measure of performance; the one that achieves the best score is

Model	Degrees of freedom
HLDA	$p+(h-1)d_0+(h-1)d_0(d_0+1)/2+p(p+1)/2$
LAD	$p+(h-1)d_0+(h-1)d_0(d_0+1)/2+p(p+1)/2+d_0(p-d_0)$
LAD2	$p+(h-1)d_0+(h-1)d_0(d_0+1)/2+p(p+1)/2+d_0(u_0-d_0)$

Table 4.1: *Degrees of freedom for computation of semi-orthogonal basis matrix* $\boldsymbol{\rho} \in \mathbb{R}^{p\times d_0}$ *for HLDA, LAD, and LAD2 methods.*

picked as *the* dimension of the reduction process. Cross-validation estimation of prediction error rates is probably the best known alternative for classification [50]. We can also rely on some of these methods for choosing d. Nevertheless, the likelihood-based approach of the methods discussed in this work allows us to use other principled methods for choosing d. Some of them can be a much less expensive alternative to cross validation. In the following paragraphs we review dimension selection methods based on likelihood-ratio statistics, simple information criteria, and permutation tests.

4.5.1 Likelihood ratio tests

The hypothesis $d = d_0$ in HLDA and LAD can be tested using the likelihood ratio statistic $\Lambda(d_0) = 2(\hat{\mathcal{L}}_p - \hat{\mathcal{L}}_{d_0})$. Here, $\hat{\mathcal{L}}_p$ is the value of the log likelihood for the considered model when using the whole set of features and $\hat{\mathcal{L}}_{d_0}$ is the log likelihood at the MLE retaining d_0 directions under the same model. Let $g(d_0)$ be a function that gives the degrees of freedom in obtaining the MLE under the considered model when looking for a dimension reduction subspace of dimension d_0. Under the null hypothesis $\Lambda(d_0)$ is distributed asymptotically as a χ^2 distribution with $g(p) - g(d_0)$ degrees of freedom. This statistic can be used to sequentially test for $d = d_0$. Starting at $d_0 = 0$ and using always the same level α for the test, the estimated dimension $\widehat{d}$ is the first hypothetized value of d_0 that it is not rejected.

The first two rows of Table 4.1 give $g(d_0)$ for HLDA and LAD. Though $g(d_0)$ can be computed formally for each model, we can explain its terms easily. For HLDA, for example, we have p parameters for the computation of the sample mean $\boldsymbol{\mu}$; $(h-1)d_0$ for the computation of traslated means $(\boldsymbol{\mu} - \boldsymbol{\mu}_y)$ for $y =$

$1, 2, \ldots, h$; $hd_0(d_0+1)/2$ for estimation of $\boldsymbol{\Omega}_y$, $(p-d_0)(p-d_0+1)/2$ for the estimation of $\boldsymbol{\Omega}_0$ and $d_0(p-d_0)$ from the fact that $\boldsymbol{\rho}_{\text{HLDA}} \in \mathbb{R}^{p\times d_0}$ lies in the Grassmann manifold of dimension d_0 in $\mathbb{R}^p$ when considering a semi-orthogonal basis matrix. Rearranging terms, we get the degrees of freedom shown in the table. In computing the degrees of freedom is important to note that setting orthogonality constraints on the projection matrix avoids estimating $\boldsymbol{\rho}_0$; it is just computed as the orthogonal complement for $\boldsymbol{\rho}_{\text{HLDA}}$. To the best of our knowledge, this simple fact has not been used in previous implementations of HLDA.

For $\boldsymbol{\rho}_{\text{LAD2}} = \boldsymbol{\rho}\mathbf{A}$ as in Section 4.4.4, a joint hypothesis $d = d_0, u = u_0$ can be tested by using the likelihood ratio statistic $\Lambda(d_0, u_0) = 2\{\mathcal{L}_{\text{full}} - \mathcal{L}(d_0, u_0)\}$, where $\mathcal{L}_{\text{full}}$ denotes the value of the maximized log likelihood for the full model and $\mathcal{L}(d_0, u_0) = \mathcal{L}(\widehat{\boldsymbol{\rho}}|d_0, u_0)$ is the maximum value of the log likelihood (4.9) for model (4.8). Under the null hypothesis, $\Lambda(d_0, u_0)$ is distributed asymptotically as a χ^2 random variable with $g(p,p) - g(u_0, d_0)$ degrees of freedom, with $g(u,d)$ given in the last row of Table 4.1. When there is only one dimension involved, it is standard practice to use a sequence of hypothesis tests to aid in its selection, as we did in HLDA and LAD before. However, in this case there seems no natural way to order the pairs (d_0, u_0) for a sequence of such tests. One way to proceed is to compare model (4.7) to the full model using the likelihood ratio statistic $\Lambda(u_0) = 2\{\mathcal{L}_{\text{full}} - \mathcal{L}(u_0)\}$, where $\mathcal{L}(u_0) = \mathcal{L}(\widehat{\boldsymbol{\rho}}|u_0)$ is the maximum value of (4.3). Under the null hypothesis $\Lambda(u_0)$ has an asymptotic χ^2 distribution with the same degrees of freedom that in the LRT for HLDA. Once again, testing is done sequentially, starting with $u_0 = 0$ and estimating u as the first hypothesized value that is not rejected. Having chosen an estimate $\hat{u}$, d can be estimated similarly treating $\hat{u}$ as known and using the likelihood ratio statistic $\Lambda(d_0, \hat{u})$ for $0 \leq d_0 \leq \hat{u}$. This method is inconsistent since there is a non-zero probability that the estimates of d and u will exceed their population values asymptotically. This probability depends on the levels of the tests. We do not regard mild overestimation of d or u as a serious issue and, in any event, overestimation in this context is a lesser problem than underestimation.

4.5.2 Information criteria

Simple information criteria like Akaike's information criterion (AIC) and Bayes information criterion (BIC) can also be used to find an estimate $\widehat{d}$ of the dimension of the central subspace. We can state both methods simultaneously. For $d_0 = 0, 1, \ldots, p$ the selected dimension for HLDA or LAD is

$$\widehat{d} = \arg\min_{d_0}\{IC(d_0) = -2\hat{\mathcal{L}}(d_0) + h(N)g(d_0)\} \ , \tag{4.13}$$

where N is the size of the sample, $h(N) = \log(N)$ for BIC, $h(N) = 2$ for AIC, and $g(d_0)$ is the same as for likelihood-ratio tests.

For the LAD2 method, dimension selection is completely analogous, just that both u and d are selected to minimize the information criterion $IC(d_0, u_0) = -2\,\mathcal{L}_{\text{LAD2}}(d_0, u_0) + h(N)g(d_0, u_0)$, with $g(d_0, u_0)$ as given in the last row of Table 4.1 and $h(N)$ as defined above for AIC and BIC.

4.5.3 Permutation tests

We can make inference on d by comparing the test statistic $\Lambda(d_0) = 2(\hat{\mathcal{L}}_p - \hat{\mathcal{L}}_{d_0})$ defined previously for LRT to its permutation distribution rather than a chi-squared distribution [23]. This allows us to get a better estimation of d when assumptions are not completely accurate. For $d_0 = 0, 1, \ldots, p-1$, a permutation distribution for $\Lambda(d_0)$ is constructed sequentially using a number P of random permutations of the sample. The observed statistic $\tilde{\Lambda}(d_0)$ is then compared to this distribution to obtain a sequence of p-values for each dimension d_0. The smallest d_0 that gives a p-value smaller than the test level α is taken to be $\hat{d}$. Though this method can give accurate inference on d for a large number of permutations of the sample, the computational load can be even harder than with cross-validation.

4.6 Experiments

In this section we use simulations to illustrate that LAD gives a better solution than HLDA for normally distributed data when covariance matrices have no special structure. We show that when data is distributed as in the HLDA model, dimension reduction using LAD is as good as using HLDA, but for more general data LAD usually needs a smaller subspace than HLDA to retain all the class-specific information. We also illustrate the equivariance of LAD under full-rank transformation of the features and the lack of this property for HLDA. We exclude LDA from the analysis as the constant covariance assumption is usually too restrictive in practice.

Throughout these experiments we work with semi-orthogonal projection matrices and use optimization over the Grassmann manifold to compute their estimators [61]. Despite this is the usual practice for LAD, it is not for HLDA for which unconstrained optimization is typically used [56]. We checked that our implementation estimates a basis matrix for the same reduction subspace than the code in [56] by verifying that the angle between the subspaces spanned by both estimates is zero [35]. Our implementation seems to require a smaller number of iterations until convergence. Nevertheless, neither of the codes are highly optimized to allow for rigurous comparison of efficiency. More details on the code used here can be found in [22].

4.6.1 HLDA vs LAD when d is known

Consider a three-class classification task and assume the data is normally distributed as $\mathbf{X}|(Y=y) \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Delta}_y)$ with

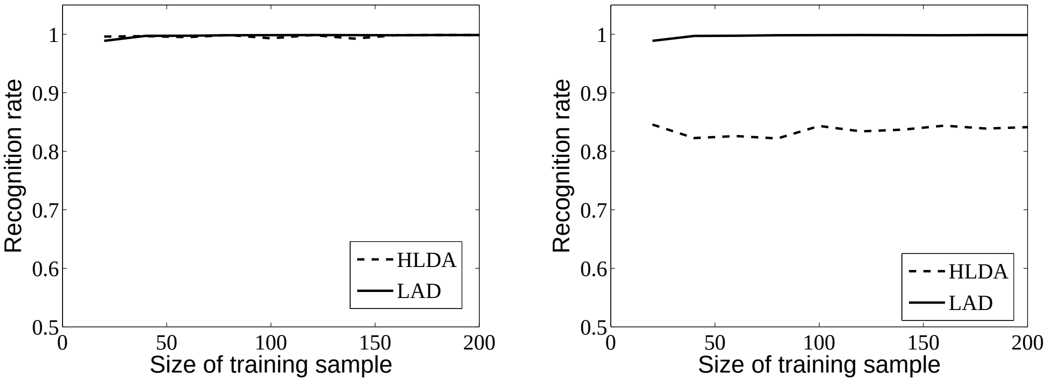
$$\begin{aligned} \boldsymbol{\mu}_y &= \boldsymbol{\rho}(\boldsymbol{\nu}_y - \bar{\boldsymbol{\nu}}_y), \\ \boldsymbol{\Delta}_y &= \boldsymbol{\Delta} + \boldsymbol{\Delta}\boldsymbol{\rho}(\boldsymbol{\Omega}_y - \boldsymbol{\Omega})\boldsymbol{\rho}^T\boldsymbol{\Delta}, \end{aligned}$$

for $y = 1,2,3$. Taking $\boldsymbol{\Delta} = \boldsymbol{\rho}\boldsymbol{\Omega}\boldsymbol{\rho}' + \boldsymbol{\rho}_0\boldsymbol{\Omega}_0\boldsymbol{\rho}_0'$, this simulation model satisfies (4.5) and the HLDA constraints.

We first ran a simulation to compare the estimates of $\boldsymbol{\rho}$ obtained by the two methods assuming we know the dimension d of the subspace spanned by $\boldsymbol{\rho}$. We took $d = 2$, $p = 10$ and choose $\boldsymbol{\nu}_1 = (1,-8)^T$, $\boldsymbol{\nu}_2 = (4,4)^T$, $\boldsymbol{\nu}_3 = (6,-7)^T$ for the projected means. For the projected covariances, we took

$$\boldsymbol{\Omega}_1 = \begin{pmatrix} 3.00 & 0.25 \\ 0.25 & 1.00 \end{pmatrix} \qquad \boldsymbol{\Omega}_2 = \begin{pmatrix} 2.0 & 0.10 \\ 0.1 & 5.00 \end{pmatrix} \qquad \boldsymbol{\Omega}_3 = \begin{pmatrix} 1.00 & -0.25 \\ -0.25 & 1.00 \end{pmatrix},$$

and we fixed a diagonal covariance matrix of dimension $(p-d)\times(p-d)$ as $\boldsymbol{\Omega}_0$. We used these models to generate samples with different sizes. For each sample



(a)

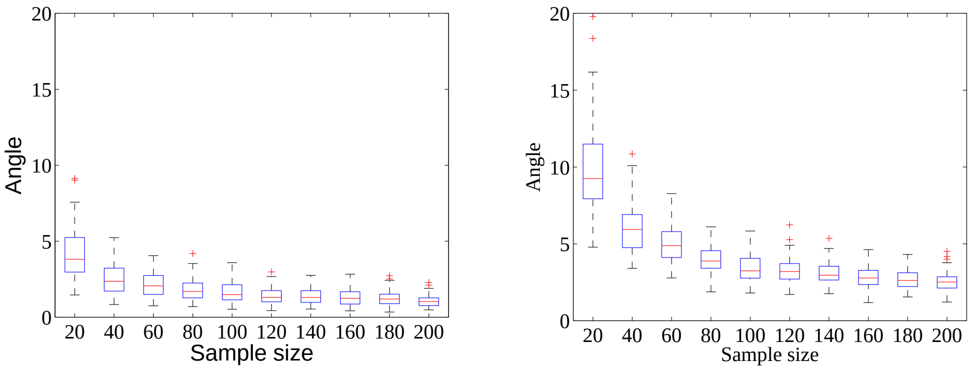
(b)

Figure 4.1: *Recognition rates for a standard quadratic classifier acting on the projected features obtained with HLDA and LAD. (a) When projecting the original observations $\boldsymbol{\mathcal{X}}$; (b) when projecting the transformed observations $\boldsymbol{\mathcal{X}}\boldsymbol{\eta}$. Classification is carried out on independent testing sets $\boldsymbol{\mathcal{X}}_T$ and $\boldsymbol{\mathcal{X}}_T\boldsymbol{\eta}$, respectively.*

size, we generated 100 replicates of a learning set $\boldsymbol{\mathcal{X}}$ and an independent equally sized testing set $\boldsymbol{\mathcal{X}}_T$. For each replicate, we computed $\boldsymbol{\rho}_{\mathrm{HLDA}}$ and $\boldsymbol{\rho}_{\mathrm{LAD}}$ using the learning set and assessed these estimates over the testing set. We first compared the recognition rates achieved with a standard quadratic classifier acting on the reduced subspace spanned by these estimates; that is, using $\boldsymbol{\mathcal{X}}_T\boldsymbol{\rho}_{\mathrm{HLDA}}$ and $\boldsymbol{\mathcal{X}}_T\boldsymbol{\rho}_{\mathrm{LAD}}$ as features. The obtained averaged recognition rates are shown in Figure 4.1-a). It is clearly seen that both estimators achieve the same performance. Only for very small sample sizes the projection with HLDA outperforms that with LAD, as it is expected *a priori* from the data generation model, but even this difference is very small.

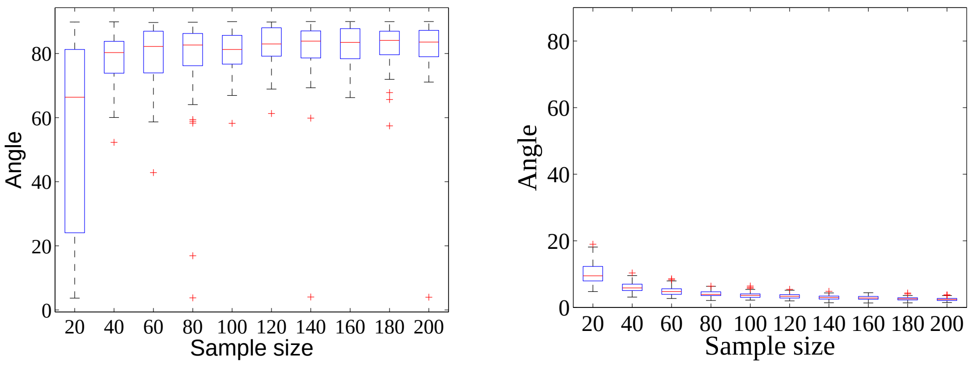
Now consider the same experiment, but with the same data multiplied by an arbitrary nonsingular matrix $\boldsymbol{\eta} \in \mathbb{R}^{p\times p}$. Obtained averaged recognition rates are shown in Figure 4.1-b). It is clearly seen that using LAD for dimension reduction leads to the same results obtained before. However, the classifier acting on the data projected with $\boldsymbol{\rho}_{\mathrm{HLDA}}$ now achieves a significantly poorer performance.

To get further insight into this example, we measured how close these estimates were to $\boldsymbol{\rho}$ by computing the angle between the projected data $\boldsymbol{\mathcal{X}}_T\boldsymbol{\rho}$ and the estimates $\boldsymbol{\mathcal{X}}_T\boldsymbol{\rho}_{\mathrm{HLDA}}$ and $\boldsymbol{\mathcal{X}}_T\boldsymbol{\rho}_{\mathrm{LAD}}$ for each replicate [35]. Figure 4.2 summarizes the obtained results. It can be seen that $\boldsymbol{\rho}_{\mathrm{HLDA}}$ is closer to $\boldsymbol{\rho}$ as it is expected,



(a) (b)

Figure 4.2: *Angle between $\boldsymbol{\mathcal{X}}_T\boldsymbol{\rho}$ and its estimates. a) Using HLDA; b) using LAD.*



(a) (b)

Figure 4.3: *Angle between $\boldsymbol{\mathcal{X}}_T\boldsymbol{\eta}\boldsymbol{\rho}$ and its estimates after transformation of the original predictors with a nonsingular matrix $\boldsymbol{\eta}$. a) Using HLDA; b) Using LAD.*

given it is a more parsimonious model for the structure of the generated data. Nevertheless, the improvement over $\boldsymbol{\rho}_{\text{LAD}}$ is important only for small sample sizes. For $N_y > 100$ it is seen that the angles obtained by both estimates remain close by around 2°. Furthermore, boxplots show the variance of the estimates is roughly the same provided the learning sample is not very small.

Now consider the same data, but multiplied by an arbitrary nonsingular matrix $\boldsymbol{\eta} \in \mathbb{R}^{p \times p}$ as before. The angles between projected transformed data $\boldsymbol{\mathcal{X}}_T\boldsymbol{\eta}\boldsymbol{\rho}$ and $\boldsymbol{\mathcal{X}}_T\boldsymbol{\eta}\boldsymbol{\rho}_{\text{LAD}}$ and between $\boldsymbol{\mathcal{X}}_T\boldsymbol{\eta}\boldsymbol{\rho}$ and $\boldsymbol{\mathcal{X}}_T\boldsymbol{\eta}\boldsymbol{\rho}_{\text{HLDA}}$ are shown in Figure 4.3. Whereas angles obtained with LAD are roughly the same as before, those obtained with

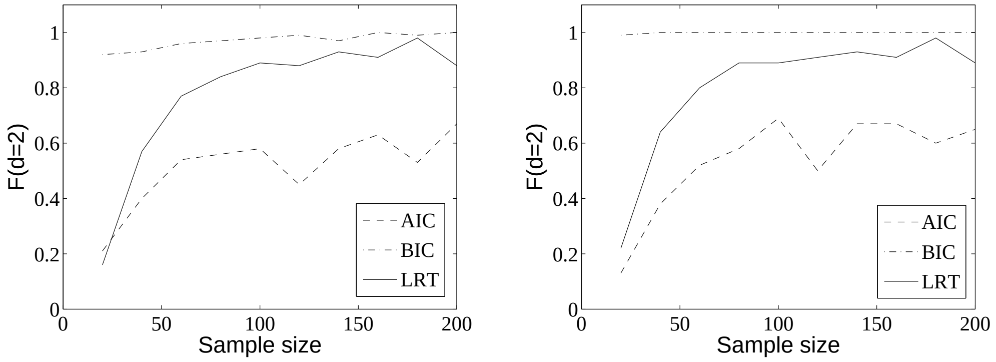
HLDA are close to 90°, which shows that $\boldsymbol{\rho}_{\text{HLDA}}$ is no longer close to $\boldsymbol{\rho}$. Indeed, the results show that there remains much information in the data that is not captured by $\boldsymbol{\rho}_{\text{HLDA}}$. This explains the drop in recognition rates obtained for the HLDA projections in Figure 4.1-b).

It is important to note that after transforming the original data with $\boldsymbol{\eta}$, the covariance matrices are no longer structured as in HLDA. Thus, the latter example also illustrates the performance of HLDA and LAD when data is normally distributed but with an arbitrary covariance matrix.

4.6.2 Inference on the dimension of the sufficient subspace

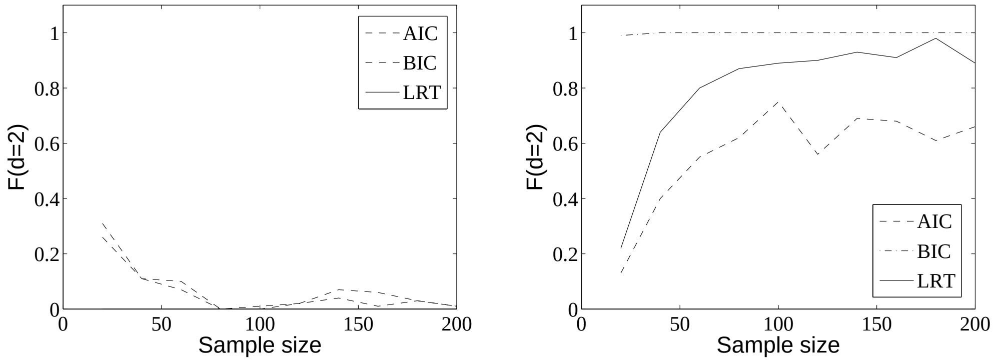
We now take the simulation set up of the previous subsection to assess the methods stated in Section 4.5 to infer about the dimension d of the minimal sufficient reduction. We know that for these data the right choice is $d = 2$. Figure 4.4 shows the fraction $F(\widehat{d} = 2)$ of the runs in which the dimension $\widehat{d}$ chosen with these methods is actually 2 as a function of sample size. We see that the different criteria perform very similarly for LAD and HLDA. Inference using BIC is found remarkably accurate, and much better than the choice given by AIC. In addition, using a test level of 5%, LRT improves when the sample size increases giving right choices more than 90% of the times when $N_y > 100$ in this example. Recall that the importance of LRT relies on the fact that it is a sequential testing procedure that avoids assessing reductions for all possible dimensions before picking the best choice for d.

We can also use these tools for infering about d to illustrate the minimality issue with HLDA. We saw above that after multiplying the data with a matrix $\boldsymbol{\eta}$ the angle between the subspace spanned by the true projection matrix $\boldsymbol{\rho}$ and the estimate $\boldsymbol{\rho}_{\text{HLDA}}$ increased and that recognition rate dropped. Figure 4.5 shows now what the fraction $F(\widehat{d} = 2)$ is for both LAD and HLDA projections of the transformed features $\boldsymbol{\mathcal{X}}\boldsymbol{\eta}$. Again, the results obtained with LAD are the same as those shown previously for the untransformed data. However, the fraction of the times that a dimension $\widehat{d} = 2$ is chosen using HLDA projections is now much smaller than before. Even more, for the transformed data $\boldsymbol{\mathcal{X}}\boldsymbol{\eta}\boldsymbol{\rho}_{\text{HLDA}}$ this fraction decreases for AIC and BIC as more observations are available to estimate $\boldsymbol{\rho}_{\text{HLDA}}$. For LRT at a 5% level, this fraction is zero for all sizes of the training



(a) (b)

Figure 4.4: *Inference on the dimension of the smallest dimension reduction subspace: a) using HLDA; b) using LAD. Figures show the fraction $F(\widehat{d} = 2)$ of the runs in which the right dimension $\widehat{d} = 2$ is chosen as the dimension of the central subspace.*



(a) (b)

Figure 4.5: *Inference on the dimension of the smallest dimension reduction subspace after re-scaling the features with a matrix $\boldsymbol{\eta}$. a) Using HLDA; b) using LAD. Figures show the fraction $F(\widehat{d} = 2)$ of the runs in which a dimension $\widehat{d} = 2$ is chosen as the dimension of the central subspace.*

sample. This strongly suggests that the subspace that retains all the class-specific information has a dimension different from $d = 2$ when constrained to the HLDA model. To find out what the chosen dimension was in these cases, we carried out a ten-fold cross validation experiment for the sample of size $N_y = 100$ to infer about d based on the minimum classification error estimate as a function of d. The method selected $\widehat{d} = 9$ in 46% of the times, $\widehat{d} = 3$ in 42% of the times,

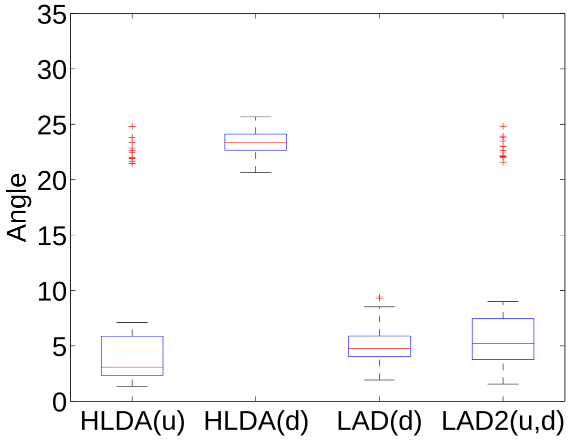
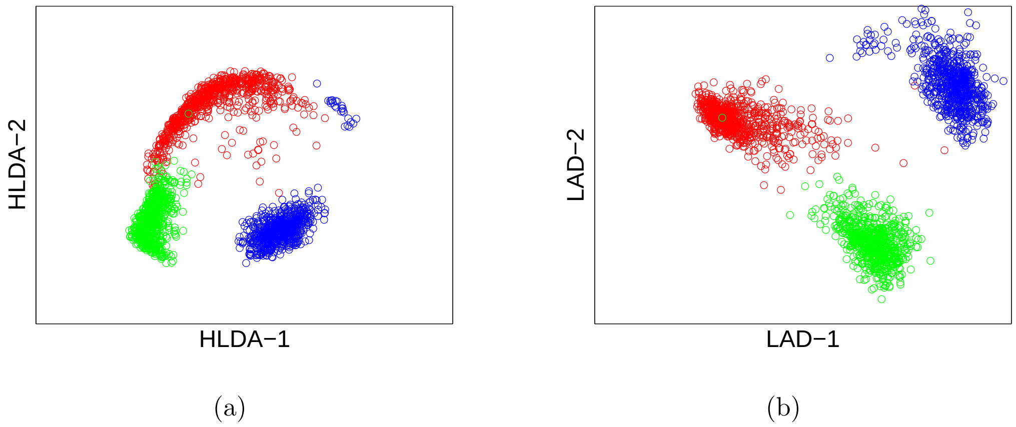


Figure 4.6: *Angle between the central subspace and several estimates for a sample of size $N_y = 500$. Boxplots were constructed after 100 runs of the experiment, using data with covariance structure as imposed in HLDA but that allows for further reduction according to (4.9).*

and the rest spread over different choices for d. As the same selection method chooses $\widehat{d} = 2$ in all the times for the original features, it becomes clear that after a simple linear transformation HLDA needs more directions to retain the class information. On the other hand, LAD continues on needing the same number of directions to do it.

4.6.3 The minimality issue revisited

To further study the lack of minimality of the HLDA estimate and compare it to LAD and the correction proposed in Section 4.4.3, we carried out another simulation using data generated from a model that has the covariance constraint of HLDA but allows for a further reduction according to (4.9). For this study we took $p = 20$, $u = 3$ and $d = 1$, defined $\boldsymbol{\rho} \in \mathbb{R}^{p \times u}$ and $\boldsymbol{\alpha} \in \mathbb{R}^{p \times d}$ and obtained $\mathbf{A} = \boldsymbol{\rho}^T \boldsymbol{\alpha}$. The central subspace is span$(\boldsymbol{\alpha})$. Figure 4.6 shows obtained angles between the central subspace and several estimates: $\boldsymbol{\rho}_{\text{HLDA}} \in \mathbb{R}^{p \times u}$, $\boldsymbol{\rho}_{\text{HLDA}} \in \mathbb{R}^{p \times d}$, $\boldsymbol{\rho}_{\text{LAD}} \in \mathbb{R}^{p \times d}$, and $\boldsymbol{\rho}_{\text{LAD2}} \in \mathbb{R}^{p \times d}$. These estimates are referred to as HLDA$_u$, HLDA$_d$, LAD$_d$ and LAD2$_{u,d}$ in the figure, respectively. This figure corresponds to 100 replicates of the experiment, using a sample size of 500 observations per class.



(a) (b)

Figure 4.7: *Linear projection of pen-digits data to a two-dimensional subspace. a) Using HLDA; b) using LAD.*

It is seen that $\boldsymbol{\rho}_{\text{HLDA}} \in \mathbb{R}^{p\times u}$ is closer to the central subspace than all of the other methods. This is not a surprise because it assumes the exact structure of covariance matrices and contains the central subspace in the population. However, this reduction retains three directions to use as features. On the other hand, the rest of the estimators retain only one transformed feature. Between them, it is seen that $\boldsymbol{\rho}_{\text{HLDA}} \in \mathbb{R}^{p\times d}$ clearly fails to span the central subspace. Nevertheless, both $\boldsymbol{\rho}_{\text{LAD}}$ and $\boldsymbol{\rho}_{\text{LAD2}}$ remain very close to the central subspace.

In other simulations with less observations available, $\boldsymbol{\rho}_{\text{LAD2}}$ showed a degraded performance, as also did $\boldsymbol{\rho}_{\text{HLDA}} \in \mathbb{R}^{p\times u}$ (not shown). Boxplots of the angles between the central subspace and these estimates becomes larger, showing a greater variablity in the obtained values compared to LAD. In addition, in a few replicates the estimates for these methods correspond to local maxima of the log likelihood function. These cases appear as outliers in the shown boxplots. Further investigation is needed to find optimal initialization of the numerical algorithm.

4.6.4 Pen digits data

Let us take some real data to further illustrate the different performances of LAD and HLDA. Consider the pen digits dataset from the UCI machine learning repository[3]. The sample was taken from 44 subjects, who were asked to write 250 random digits. Using standard preprocessing techniques, each written digit yields a 16-dimensional feature vector which is used for classification. The 44 subjects were divided into two groups of size 30 and 14, in which the first formed the training set and the second formed the test set. Figure 4.7 illustrate dimension reduction of the feature vectors from the training set to a subspace of dimension $d = 2$. This transformation would serve as a preparatory step for developing the classifier. For clarity, we only took the digits 0, 6 and 9, which reduced the sample to 2,219 cases. This subset has also been considered previously for illustration purposes [97]. The data projected using LAD results in separate clusters for each class, which could be well-modeled using Gaussian distributions. HLDA projections, on the other hand, show a worse defined distribution and some overlap over the classes. The different quality of these reductions impact on the performance of the classifier. Using a standard quadratic classifier on the two-dimensional subspace of the projected features, the error rate with HLDA projections is 5%. Using LAD projections instead of HLDA projections, the error rate reduces 60% down to 2%. To get an error rate close to that for LAD for this dataset, HLDA needs to retain four directions instead of two.

4.7 Concluding remarks

In this chapter, we have focused on information retention when using likelihood-based methods for dimension reduction of normally distributed data. LDA and HLDA have been analyzed under the framework of likelihood-based sufficient dimension reduction and conditions on the data have been stated in order to allow these methods to retain all the class information. It has been shown that HLDA often needs to retain more directions than the strictly necessary, to account not only for all the class information but also to satistfy the assumed structure in the covariance matrices. On the other hand, it has been shown that the LAD estimator provides a better solution for subspace projection of heteroscedastic data without constraints, giving a reduction that is minimal and satisfies an important invariance property. In addition, a new estimator LAD2 was introduced to deal with data that actually have a structured covariance matrix as assumed

[3] ftp://ftp.ics.uci.edu/pub/machine-learning-databases/pendigits

in HLDA. Unlike HLDA, however, the proposed method guarantees minimal reductions and it is more efficient than LAD for this type of data. Understanding existing methods under sufficiency has allowed us to state inference methods about the dimension of the smallest reduction subspace that is sufficient to retain class information. This interpretation has led also to new implementations of the existing methods using matrix orthogonality constraints that seem to improve computational efficiency and avoids explicit computation of the rejected non-discriminant subspace.

CHAPTER 5

SDR for hidden Markov models

5.1 Introduction

In Chapter 4 we discussed likelihood-based dimension reduction methods for normal models. Using theory from the sufficiency framework, we gave theoretical insight about information retention with projection methods commonly used in HMM-based pattern recognition, such as LDA and HLDA. We discussed optimality of these methods in avoiding information loss and, most important, we showed that LAD is a better way than HLDA to deal with heteroscedastic data, as it does not impose restrictive constraints on the covariance and usually needs a smaller number of linear combinations of the original features to retain all the information. It turns out that SDR for condional normal models is all we need to use the SDR methodology for dimension reduction of GHMM. As exploited previously in [57, 56, 83], the connection relies in using the EM algorithm for maximum likelihood parameter estimation.

In this chapter we derive a SDR method for HMM with Gaussian observation densities and propose a simplified algorithm for practical implementation. The proposed method is compared against HLDA, which is the most widely used alternative for dimension reduction for HMM.

5.2 Derivation of the proposed method

Assume a HMM ϑ_i is used to model sequences of random vectors $\mathbf{X}^n = \{\mathbf{x}_t^n\}$ from the i-th class, with $\mathbf{x}_t^n$ be a realization of a random variable X_t. Let Q_t be the state of the underlying Markov chain at time t, and assume that Q_t takes one out of N_i values $q = 1, 2, \ldots, N_i$. In addition, assume ϑ_i is homogeneous and has Gaussian observation densities. This means that conditional on the state Q_t, the random vector of features X_t is normally distributed as $\mathrm{p}(X_t|Q_t = q, \vartheta_i) = \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Delta}_q)$. In this scenario, $\boldsymbol{\rho}$ is a basis matrix for a dimension reduction subspace if

$$X_t|(\boldsymbol{\rho}^T X_t, Q_t = q, \vartheta_i) \sim X_t|\boldsymbol{\rho}^T X_t. \tag{5.1}$$

In this way, $\boldsymbol{\rho}^T X_t$ and X_t have the same information on the state Q_t of the model for class i, for every q and i. Thus, if we map (q, ϑ_i) onto a single index $y_{q,i}$, we recover the condition for SDR of normal data as discussed in Chapter 4.

Nevertheless, for each random vector of features X_t, the only label that is available is the class i, as Q_t remains hidden to the observer. Thus, SDR methods for normal data cannot be used straightforwardly in this context. As a first approach to get labels at the state level for each random vector X_t, each HMM can be trained first with the standard training algorithm using the original features. In a second step, these trained models can be used to make inference about the optimal sequences of states to describe the feature vectors in the training set, and the inferred states can be used as labels for the observed features. As observations are assumed normally distributed given the state of the Markov chain, LAD can be applied over this labeled dataset to obtain a basis matrix $\widehat{\boldsymbol{\rho}}$ for the dimension reduction subspace. Once $\widehat{\boldsymbol{\rho}}$ has been obtained, parameters for observation densities in the HMM can be transformed using $\widehat{\boldsymbol{\rho}}$ and classification can be carried out in the reduced subspace after projection of the testing set with $\widehat{\boldsymbol{\rho}}$. This training scheme for the classifier will be used later for comparison and it will be referred to as $\mathrm{HMM}_{\text{EXT-LAD}}$.

Despite the procedure described above being appealing to generate a labeled dataset to apply the SDR methodology, it does not consider the central subspace properly as another parameter to estimate. A better solution is to embed the dimension reduction task into the Baum-Welch algorithm for estimation of the full model parameters [4, 30]. In this case, to obtain a maximum likelihood estimate of a semi-orthogonal basis matrix $\boldsymbol{\rho}$ of the dimension reduction subspace, we should maximize the expectation

$$\mathcal{Q}(\boldsymbol{\rho};\{\vartheta_i\}) = \frac{1}{2}\sum_{i=1}^{h}\sum_{q=1}^{N_i}\sum_{\mathbf{X}^k\in\mathcal{X}_i}\sum_{t=1}^{T_k}\gamma_t^k(y_{q,i})\log\frac{|\boldsymbol{\rho}^T\tilde{\boldsymbol{\Sigma}}\boldsymbol{\rho}|}{|\boldsymbol{\rho}^T\tilde{\boldsymbol{\Delta}}_{y_{q,i}}\boldsymbol{\rho}|}, \tag{5.2}$$

where $\mathcal{X}_i$ denotes the training set for class i, comprising sequences of correlated observations $\mathbf{X}^k = \{\mathbf{x}_{k1}, \mathbf{x}_{k2}\ldots, \mathbf{x}_{kT_k}\}$, $\gamma_t^k(y_{q,i}) = \mathrm{p}(Q_t = q|\mathbf{X}^k, \vartheta_i)$, and the number of observations T_k may be different for each sequence $\mathbf{X}^k$. In addition,

$$\tilde{\boldsymbol{\mu}}_{y_{q,i}} = \frac{\sum_{\mathbf{X}^k\in\mathcal{X}_i}\sum_{t=1}^{T_k}\gamma_t^k(y_{q,i})\mathbf{x}_t^k}{\sum_{\mathbf{X}_k\in\mathcal{X}_i}\sum_{t=1}^{T_k}\gamma_t^k(y_{q,i})}, \tag{5.3}$$

$$\tilde{\boldsymbol{\mu}} = \frac{\sum_{i=1}^{h}\sum_{q=1}^{N_i}\sum_{\mathbf{X}^k\in\mathcal{X}_i}\sum_{t=1}^{T_k}\gamma_t^k(y_{q,i})\mathbf{x}_t^k}{\sum_{i=1}^{h}\sum_{q=1}^{N_i}\sum_{\mathbf{X}^k\in\mathcal{X}_i}\sum_{t=1}^{T_k}\gamma_t^k(y_{q,i})}, \tag{5.4}$$

$$\tilde{\boldsymbol{\Delta}}_{y_{q,i}} = \frac{\sum_{\mathbf{X}^k\in\mathcal{X}_i}\sum_{t=1}^{T_k}\gamma_t^k(y_{q,i})(\mathbf{x}_t^k-\tilde{\boldsymbol{\mu}}_{y_{q,i}})(\mathbf{x}_t^k-\tilde{\boldsymbol{\mu}}_{y_{q,i}})^T}{\sum_{\mathbf{X}^k\in\mathcal{X}_i}\sum_{t=1}^{T_k}\gamma_t^k(y_{q,i})}, \tag{5.5}$$

$$\tilde{\boldsymbol{\Sigma}} = \frac{\sum_{i=1}^{h}\sum_{q=1}^{N_i}\sum_{\mathbf{X}^k\in\mathcal{X}_i}\sum_{t=1}^{T_k}\gamma_t^k(y_{q,i})(\mathbf{X}_t^k-\tilde{\boldsymbol{\mu}})(\mathbf{X}_t^k-\tilde{\boldsymbol{\mu}})^T}{\sum_{i=1}^{h}\sum_{q=1}^{N_i}\sum_{\mathbf{X}^k\in\mathcal{X}_i}\sum_{t=1}^{T_k}\gamma_t^k(y_{q,i})}. \tag{5.6}$$

Note that (5.2) is analogous to the log-likelihood function (4.6) for the LAD method. Nevertheless, here we do not have certainty about the true normal population from where the observations come. Instead, quantities $\gamma_t^k(y_{q,i})$ estimate the posterior probability of the random vector $\mathbf{x}_t^k$ from the sequence $\mathbf{X}^k \in \mathcal{X}_i$ of coming from the normal observation model indexed by $y_{q,i}$. Thus, the sums $\sum_{\mathbf{X}^k\in\mathcal{X}_i}\sum_{t=1}^{T_k}\gamma_t^k(y_{q,i})$ have an analogous meaning to N_y in (4.6). The posterior probabilities $\gamma_t^k(y_{q,i})$ are computed for each training sample and at each iteration of the algorithm, using the current estimates of the parameters of model ϑ_i. In this way, the statistics (8)-(11) vary also at each iteration of the algorithm. It is important to stress that the different populations for the dimension reduction task are all the conditional observation models in each HMM. Thus, the dimension reduction task will involve $\sum_i N_i$ normal populations.

5.2.1 Simplified algorithm

As a simpler alternative to the training approach described above, we can use Viterbi's algorithm to make inference on the optimal sequences of hidden states

Algorithm 1: Joint estimation of the reduction and model parameters

```
• Initialization
    1. For each class $i$, set $\vartheta_i^* = \vartheta_i$ and $\mathcal{X}_i^* = \mathcal{X}_i$.
    2. Let $\mathcal{X}^* = \bigcup_i \mathcal{X}_i^*$ be the whole training sample.
• Main loop: repeat until convergence
    1. For each class $i$, infer the optimal sequences of states $\{\mathbf{Q}_k^*\}_i$ to describe
       the data $\mathcal{X}_i^*$ according to $\vartheta_i^*$.
    2. Form the whole labeled dataset $\Upsilon = \bigcup_i (\{\mathbf{Q}_k^*\}_i, \mathcal{X}_i)$.
    3. Estimate the semiorthogonal basis matrix $\boldsymbol{\rho}^*$ for LAD using $\Upsilon$.
    4. Compute $\boldsymbol{\rho}_0^*$ that spans a subspace orthogonal to span($\boldsymbol{\rho}^*$).
    5. Build the orthogonal matrix $\boldsymbol{\Theta} = (\boldsymbol{\rho}^* \boldsymbol{\rho}_0^*)$.
    6. Lineary transform the original dataset using $\boldsymbol{\Theta}$ to obtain a new $\mathcal{X}^*$.
    7. For each class $i$, update the observation model corresponding to each estate
       $q$ of $\vartheta_i$, doing $\boldsymbol{\mu}_{y_{q,i}}^* = \boldsymbol{\Theta}^T \boldsymbol{\mu}_{y_{q,i}}$ and $\boldsymbol{\Delta}_{y_{q,i}}^* = \boldsymbol{\Theta}^T \boldsymbol{\Delta}_{y_{q,i}} \boldsymbol{\Theta}$.
• Finalization
    1. Set $\widehat{\boldsymbol{\rho}} = \boldsymbol{\rho}^*$.
    2. For each class $i$, take the first $d$ coordinates of the model parameters in $\vartheta_i^*$
       to build the final estimates of the models, $\widehat{\vartheta}_i$.
```

to describe the observed data, instead of summing up all the possible paths as in the Baum-Welch algorithm [6, 81]. Assume we have pre-initialized HMM models ϑ_i, one for each data class and let $\mathcal{X} = \bigcup_i \mathcal{X}_i$ refer to the whole training set. The proposed algorithm is shown in Figure 1. In each iteration, Viterbi's algorithm is used to infer the sequences of states $\mathbf{Q}_k^* = \{Q_{kt}^*\}$ that best describes each sequence $\mathbf{X}_k$ in the training sample $\mathcal{X}_i$, according to the corresponding HMM ϑ_i. The obtained posteriors are used to label the training sample at the hidden-state level. The LAD method is applied to the labeled dataset to estimate a basis matrix $\boldsymbol{\rho}$ for the central subspace. After computing a basis matrix $\boldsymbol{\rho}_0$ for the orthogonal complement to the estimated central subspace, the orthogonal matrix $(\boldsymbol{\rho}, \boldsymbol{\rho}_0)$ is used to lineary transform the training set $\mathcal{X}$ and the initial estimates of the parameters of the models. These steps are repeated until there is no significant increase in the likelihood of the dataset. After convergence, only the first d coordinates of the model parameters are retained. Classification is carried out in this reduced subspace, after projecting the test samples with the estimated basis matrix $\widehat{\boldsymbol{\rho}}$.

It is important to note that inference about the optimal sequences of states is carried out in a (transformed) feature space, without rejecting any coordinate. This is aimed to avoid lossing important information in the first iterations of the algorithm, when only rough estimates of model parameters are available to label the observations. As the reduction is supervised according to this label assignment and the later can be inaccurate in the beginning of the training, rejecting coordinates may imply lossing relevant information that cannot be recovered in next iterations. Furthermore, the update of both the dataset and the parameters for the observation models of the HMM are computed transforming always the original training sample and the initial estimates of model parameters, respectively. Preliminary experiments with both synthetic and real data showed that the algorithm implemented in this way is stable and it was found to reach convergence typically after a few iterations.

5.3 Experiments with synthetic data

This section describes simulation studies aimed to illustrate the main properties of the proposed dimension reduction method for HMM-based classifiers.

5.3.1 Set up

We ran a simulation study for a two-class discrimination problem. For this experiment, data for each class was generated using a corresponding HMM with Gaussian observation densities. The number of hidden states was set to three ($N_i = 3$) for both models $i = 1, 2$. Conditional on the state of the Markov chain, observed data was generated from a normal population with parameters

$$\begin{aligned}
\boldsymbol{\mu}_y &= \boldsymbol{\rho}(\boldsymbol{\nu}_y - \bar{\boldsymbol{\nu}}), \\
\boldsymbol{\Delta}_y &= \boldsymbol{\Delta} + \boldsymbol{\Delta}\boldsymbol{\rho}(\boldsymbol{\Omega}_y - \boldsymbol{\Omega})\boldsymbol{\rho}^T\boldsymbol{\Delta},
\end{aligned}$$

with $\bar{\boldsymbol{\nu}} = \sum_y \boldsymbol{\nu}_y/6$, $\boldsymbol{\Omega} = \sum_y \boldsymbol{\Omega}_y/6$ and $\boldsymbol{\Delta} = \boldsymbol{\rho}\boldsymbol{\Omega}\boldsymbol{\rho}^T + \boldsymbol{\rho}_0\boldsymbol{\Omega}_0\boldsymbol{\rho}_0^T$. Comparison of this normal model with (4.7) shows that it satisfies the conditions to make HLDA an

HMM for class 1		
$\mathbf{A}_1 = \begin{pmatrix} 0.60 & 0.35 & 0.05 \\ 0 & 0.75 & 0.25 \\ 0 & 0 & 1.00 \end{pmatrix}$		
state 1	*state 2*	*state 3*
$\boldsymbol{\nu}_{1,1} = (1, -3)^T$ $\boldsymbol{\Omega}_{1,1} = \begin{pmatrix} 1.00 & -0.25 \\ -0.25 & 3.00 \end{pmatrix}$	$\boldsymbol{\nu}_{2,1} = (4, 2)^T$ $\boldsymbol{\Omega}_{2,1} = \begin{pmatrix} 2.00 & 1.50 \\ 1.50 & 5.00 \end{pmatrix}$	$\boldsymbol{\nu}_{3,1} = (3, -1)^T$ $\boldsymbol{\Omega}_{3,1} = \begin{pmatrix} 1.00 & -0.25 \\ -0.25 & 1.00 \end{pmatrix}$
HMM for class 2		
$\mathbf{A}_2 = \begin{pmatrix} 0.75 & 0.15 & 0.10 \\ 0 & 0.75 & 0.25 \\ 0 & 0 & 1.00 \end{pmatrix}$		
state 1	*state 2*	*state 3*
$\boldsymbol{\nu}_{1,2} = (-1, 0)^T$ $\boldsymbol{\Omega}_{1,2} = \begin{pmatrix} 3.00 & 0.25 \\ 0.25 & 1.00 \end{pmatrix}$	$\boldsymbol{\nu}_{2,2} = (2, 2)^T$ $\boldsymbol{\Omega}_{2,2} = \begin{pmatrix} 2.00 & 1.50 \\ 1.50 & 5.00 \end{pmatrix}$	$\boldsymbol{\nu}_{3,2} = (2, -3)^T$ $\boldsymbol{\Omega}_{3,2} = \begin{pmatrix} 1.00 & -0.45 \\ -0.45 & 1.00 \end{pmatrix}$

Table 5.1: *HMM parameters used in the simulation. Notation $\lambda_{q,i}$ refers to the parameter λ for the normal density corresponding to state q of the HMM for class i.*

optimal method for dimension reduction. Our objective in choosing this is twofold: on the one hand, we want to show that when the data is exactly as assumed by HLDA, the reduction obtained with LAD is as good as the one obtained with HLDA. On the other hand, if this original data is lineary transformed with a nonsingular matrix $\boldsymbol{\eta}$, the covariance structure gets broken and HLDA is no longer optimal. We want to show that in this case, which also accounts for a general covariance matrix of the populations, LAD is significantly better than HLDA. Furthermore, this condition should illustrate that error rates achieved using LAD-derived estimators remains fairly the same after transforming the features, due to the equivariance property of the estimator [20].

A feature space with dimension $p = 10$ was used, with a central subspace of dimension $d = 2$. Table 5.1 shows the values set for HMM parameters in the sufficient subspace. Matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ have the probabilities of transition between different states for the HMM corresponding to class 1 and 2, respectively. That is, $(\mathbf{A}_1)_{ij}$ is the probability that the HMM for class 1 jumps from state $Q_{t-1} = j$ to $Q_t = i$ at any time t. Note that there is not anything special

Exp.	Sample size	HMM$_{\text{NORED}}$	HMM$_{\text{EXT-LAD}}$	HMM$_{\text{HLDA}}$	HMM$_{\text{LAD}}$
A	2×100	0.1465	0.0805	0.0425	0.0220
	2×1000	0.1080	0.0928	0.0424	0.0229
	2×5000	0.1949	0.1014	0.0571	0.0234
B	2×100	0.1445	0.1045	0.2045	0.0235
	2×1000	0.1153	0.0954	0.1698	0.0222
	2×5000	0.1549	0.1043	0.1925	0.0237

Table 5.2: *Error rate obtained with each classifier for different sizes of the training set. Reported values are means over ten runs of the experiment. Exp. A: data was generated using normal models as assumed in HLDA. Exp. B: data was obtained linearly transforming the data used in A.*

in the chosen values. They could have been set at random, but specific values have been prefered to make the experiment easily reproducible. A training set and an independent testing set were randomly generated for each class using the model parameters stated above. Each generated sequence $\mathbf{X}_k$ had a number T_k of feature vectors which varied randomly as $6 \leq T_k \leq 10$. Each feature vector $\mathbf{X}_{kt}$ was drawn from a Gaussian density conditional on the state of the related hidden Markov chain at that time. For the dimension reduction stage, computations were carried out using an extension to HMM of the software available from [**?**].

5.3.2 Results

The performance of the following classifiers was compared: i) HMM$_{\text{NORED}}$, in which each HMM was trained with the Baum-Welch algorithm using the original 10-dimensional feature space; ii) HMM$_{\text{EXT-LAD}}$, which includes dimension reduction using LAD but it is not embedded in HMM training; iii) HMM$_{\text{LAD}}$, in which LAD is embedded in the iterative training process described in Algorithm 1; and iv) HMM$_{\text{HLDA}}$, in which HLDA replaces LAD in Algorithm 1.

We ran the experiment for different sizes of the training set. In all the cases, classification was carried out over independent test sets with the same size as the training set used in the given experiment. The same datasets and the same initial estimates of the models were used for all the classifiers, so that random initializa-

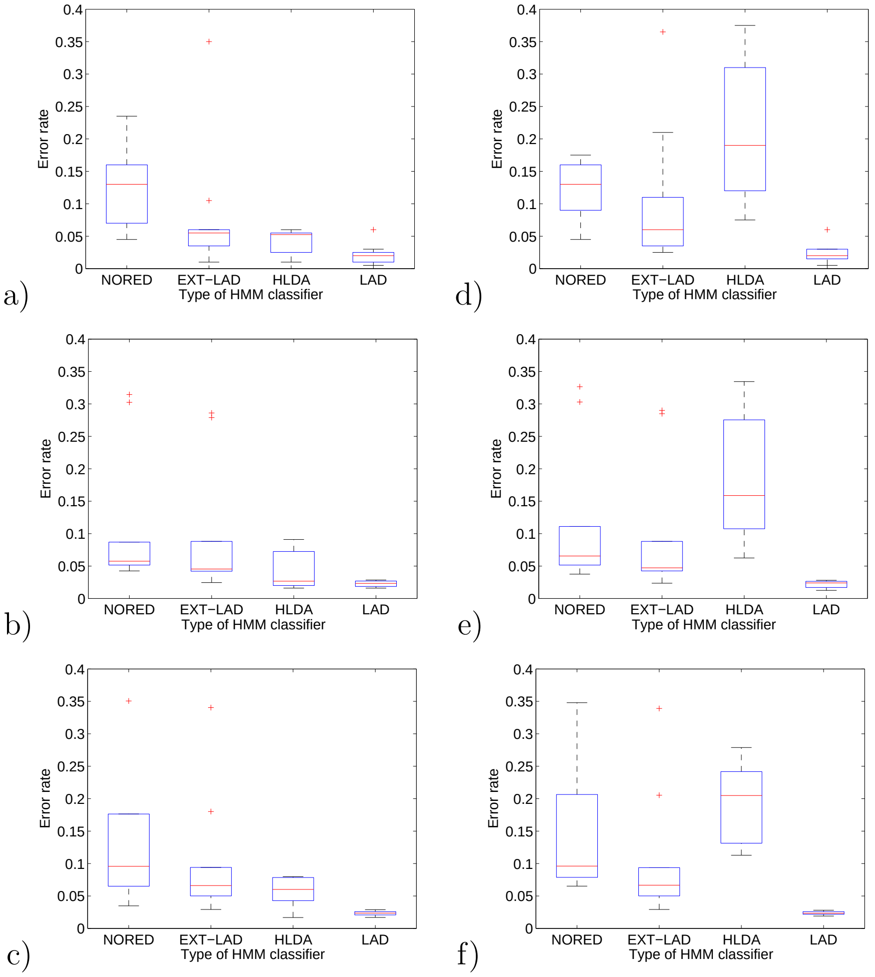


Figure 5.1: *Error rates achieved with classifiers HMM$_{\text{NORED}}$, HMM$_{\text{EXT-LAD}}$, HMM$_{\text{HLDA}}$ and HMM$_{\text{LAD}}$ for different sizes of the training and testing samples. a) 100 sequences per class; b) 1000 sequences per class; c) 5000 sequences per class; d)-f) same as a)-c), respectively, but after linear transformation of the data with a matrix $\boldsymbol{\eta}$.*

tion has no effect on the relative performance of the tested schemes. Statistical significance of the obtained mean scores were analyzed according to this random block design. Experiment A in Table 5.2 shows the obtained results. Reported error rates are mean values over ten runs. It can be seen that the embedded alternatives HMM$_{\text{HLDA}}$ and HMM$_{\text{LAD}}$ both outperform HMM$_{\text{EXT-LAD}}$ and HMM$_{\text{NORED}}$. On the one hand, these results confirm that the performance of the classifier can be improved reducing the dimensionality of the feature space. In addition, the magnitude of the improvement depends on the obtained reduction. Since the projections depend on the labels assigned to the observed data, embedding the dimension reduction into the estimation of the model parameters allows for a sinergistic interplay between the estimated projection and the labelling process from an iteration of the algorithm to the next. It is clearly seen also that HMM_{LAD} outperforms the other alternatives; results are significant at the 5% level for each size of the training set. It is interesting to note that the superiority of HMM_{LAD} is significant even for the smallest training sample. The later result is important since HLDA is a tighter model for the data by the design of the experiment. It suggests that inference is actually harder for such structured model compared to LAD. We explore this hypothesis more deeply below.

In addition to the mean scores, it is interesting to study the variability of the achieved recognition rates using the different classifiers. Boxplots in Figure 5.1-a) to c) show this information. As expected, the performance of classifiers including dimension reduction presents smaller variability compared to that of $\text{HMM}_{\text{NORED}}$. More interesting, there is not a significant difference in the variablity achieved by $\text{HMM}_{\text{EXT-LAD}}$ and HMM_{HLDA}, but results for HMM_{LAD} are much closer to their mean value. The very small variance obtained with HMM_{LAD} is found even for the smallest sample size used to train the classifier. These results suggest that the error rates achieved using HMM_{LAD} are close to the minimum attainable for the particular dataset using the common Bayes classification rule.

Finally, let us consider the effect of transforming the dataset with a nonsingular matrix $\boldsymbol{\eta} \in \mathbb{R}^{p \times p}$ generated randomly. Obtained results are given in Experiment B of Table 5.2. If we concentrate on the difference in error rate achieved with each reduction method after transforming the features compared to its performance with the original features, it is found that increments are not significant for $\text{HMM}_{\text{NORED}}$, $\text{HMM}_{\text{EXT-LAD}}$ and HMM_{LAD}, but HMM_{HLDA} is strongly affected by the transformation. As the special covariance structure of the original data no longer holds after the transformation with $\boldsymbol{\eta}$, these results clearly illustrate that for a fixed dimension of the dimension reduction subspace, HLDA is an optimal reduction only for a very particular covariance structure. Thus, more directions should have been retained using HLDA to conserve all the original information and to avoid increasing the error rate.

The corresponding boxplots shown in Figure 5.1-d) to f) further illustrate the comments above. It can be seen that boxplots remain fairly the same after transforming the features for the reduction methods involving LAD, albeit there is some increasement in variability for the non-embedded alternative when it is

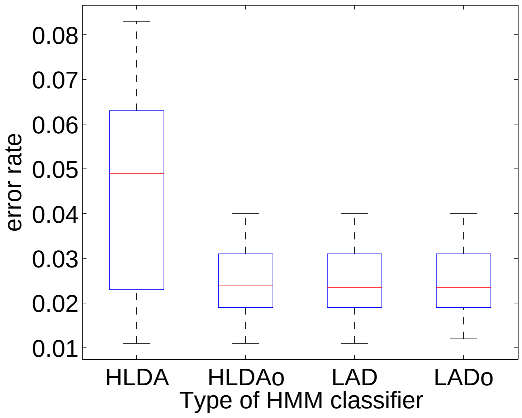


Figure 5.2: *Comparison of error rates achieved when using inference on the sequences of states that generated the observations (HMM$_{HLDA}$ and HMM$_{LAD}$), against using the true paths for labelling the data for the dimension reduction stage (HMM$_{HLDAo}$ and HMM$_{LADo}$).*

used with the smallest training sample. Unlike this, the variablity of the scores obtained using HMM$_{\text{HLDA}}$ raises remarkably after transformation of the original data, for every size of the training sample. This greater variability of the error rates is due to the different amounts of discriminant information that is present in the retained directions.

5.3.3 Why does LAD perform better even for data with the covariance structure of HLDA?

It is worth noting that the reduction method based on LAD seems to achieve significant better results than HLDA even for data generated from conditional normal models with the covariance structure assumed by HLDA. To get some insight into this effect, we ran a simulation to analyze the hypothesis that LAD allows actually for more accurate inference about the most probable sequences of states in the models to describe the observed data. This would allow for a better labelling of the data for the dimension reduction stage, which contributes to getting a better estimate of the projection, that in turn will help in the inference step at the next iteration of the algorithm. To check this, we compared the performance of HMM$_{\text{LAD}}$ and HMM$_{\text{HLDA}}$, implemented as proposed in Algorithm

1, against classifiers $\mathrm{HMM}_{\mathrm{LADo}}$ and $\mathrm{HMM}_{\mathrm{HLDAo}}$, respectively, which differ from the formers just in that the true paths that generated the synthetic data are used, instead of infering them from the data. As in previous simulations, independent training and testing samples were used in each run. Results over ten runs of the experiment are shown in Figure 5.2. It can be seen that $\mathrm{HMM}_{\mathrm{LAD}}$ achieves fairly the same results as $\mathrm{HMM}_{\mathrm{LADo}}$, meaning that inference on the most likely sequences of states is very good and does not lead to an increase in error rates compared to the exact labelling when using the true paths as in $\mathrm{HMM}_{\mathrm{LADo}}$. On the other hand, inference on the optimal sequence of states is shown to significantly reduce the performance of $\mathrm{HMM}_{\mathrm{HLDA}}$ compared to $\mathrm{HMM}_{\mathrm{HLDAo}}$. These results clearly agree with the hypothesis stated above.

5.4 Experiments with real data

A phoneme recognition experiment using data from the TIMIT speech corpus [99] was carried out to illustrate the performance of the proposed method with real data. For this task, we used samples of phonemes /b/, /d/, /eh/, /ih/ and /jh/. The voiced stops /b/ and /d/ have a very similar articulation and they have different phonetic variants according to the context. Vowels /eh/ and /ih/ were selected because their formants are very close [81]. All of this makes these pairs very confusable. The affricate phoneme /jh/ was added as representative of the voiceless group to complete the set. It is important to note that these phoneme are not spoken isolatedly, but extracted from continuous speech. Because of that, there is a large variability in both acoustic features and duration in the samples of each phoneme, contributing to a more difficult discrimination task.

Speech signals were analyzed in frames of 30 ms length, using Hamming windows with 50% overlap between consecutive frames. 12 cepstral coefficients in the Mel frequency scale [28, 81] along with a log-energy coefficient were extracted from each segment, thus giving sequences with $\mathbf{x}_t \in \mathbb{R}^{13}$. Random samples of 200 sequences per phoneme taken from the TIMIT training set were used for parameter estimation, and independent 200 sequences per phoneme from the testing set were used for classification. For each class, a HMM with $N_i = 3$ was trained, with observation densities given by a mixture of two normal densities; that is we assume $p(X_t|Q_t = q, \vartheta_i) = m_1\mathcal{N}(\boldsymbol{\mu}_{y_{q,i},1}, \boldsymbol{\Delta}_{y_{q,i},1}) + (1 - m_1)\mathcal{N}(\boldsymbol{\mu}_{y_{q,i},2}, \boldsymbol{\Delta}_{y_{q,i},2})$.

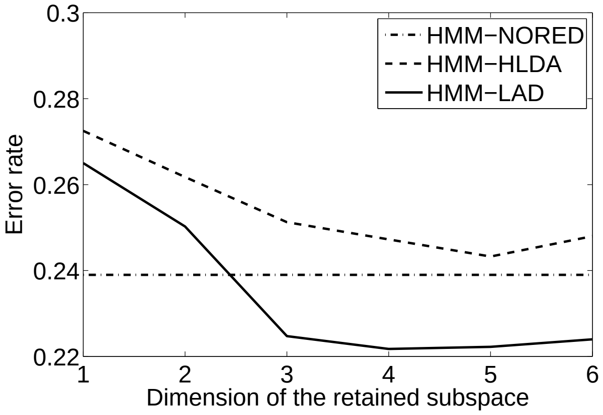


Figure 5.3: *Error rates for several dimensions of the retained subspace. Shown results are averages over ten runs of the experiment.*

State-transition matrices for each HMM were initializaed at random, and initial estimates for the Gaussian parameters were obtaiend using a standard k-means algorithm. Ten runs of the experiment were carried out. In each run, the training and test samples as well as the initial estimates for the models parameters were fixed for all the classifiers, to reduce the influence of these factors in the achieved recognition rates.

Figure 5.3 shows the obtained error rates for classifiers $\text{HMM}_{\text{NORED}}$, HMM_{HLDA} and HMM_{LAD}, as a function of the dimension of the retained subspace. Reported results are the averages over the ten runs of the experiment. It can be seen that when the retained subspace has dimension $d < 3$, both classifiers including dimension reduction perform poorer than the classifier that does not reduce the dimensionality of the data, indicating that some important discriminant information is lost if only such a few directions are retained. However, for $d \geq 3$, HMM_{LAD} performs significantly better than $\text{HMM}_{\text{NORED}}$. A one-tail paired t-test between the scores for $\text{HMM}_{\text{NORED}}$ and HMM_{LAD} for $d = 4$, dimension at which the minimum error rate is attained, shows a p-value of 0.023, meaning that the improvement in error rate is significant. On the other hand, it can be seen that the performance of HMM_{HLDA} remains significantly poorer for all the assessed dimensions of the retained subspace. In fact, for this example, using HLDA is shown to be worst than not reducing at all the dimensionality of the feature space. Clearly, HLDA needs to conserve more directions to avoid losing important features for discrimination.

5.5 Conluding remarks

The EM algorithm allows us to seamlessly integrate the SDR methods for normal models into the training process for HMM. As a result, consistent methods for SDR of HMM-modeled data are obtained. In particular, a new projection method based on LAD extends to HMM settings the better properties and theoretical background of this method as compared to HLDA. Simulations shown that the proposed method based on LAD clearly outperforms HLDA when no further structure in the covariance matrices is assumed, and that it is as good as HLDA even under the condition that are most favorable for this method. Experiments with real speech data sets showed that theoretical advantages of LAD translate also to a superior performance in applications with real data. It is important to note that the proposed method has the same computational complexity as HLDA and does not require further adaptations to replace it in current software for manipulation of HMM in large scale applications.

CHAPTER 6

Conclusions and further research

In this thesis, discriminative information in HMM-based classifiers has been addressed from two different points of view. On one hand, a new training method for HMM-HMT models was proposed, which uses information from all the classes to emphasize differences between the models in order to minimize the expected classification error rate. On the other hand, retention of discriminative information when applying linear dimension reduction in GHMM-based classifiers was analyzed using the framework of sufficient dimension reduction. In this regard, we advanced in understanding information loss when using existing methods, and new reductions for HMM that are optimal in the sense of sufficiency were proposed using results for normal populations as a building block.

The discriminative training method for HMM-HMT models introduced here extended the minimum classification error approach to sequences of data observed in the wavelet domain and modeled through HMT. An adaptation of the Viterbi algorithm was used to define the set of discriminant functions. The training algorithm also required special considerations about the HMT observation models and the feature space to derive useful measures of misclassification to approximate the decision risk of the classifier. In particular, comparing the order of magnitud of the discriminant functions was found to be better than weighting their actual values. The resulting algorithm does not only penalize confusability of the training patterns to drive the learning process, as previous methods do, but also do it with increased strength for misclassified observations. In this way, it adds a corrective actuation that it is not usual in standard settings of MCE training but proves to work well in this context.

Experiments in phoneme recognition showed that the proposed method consistently outperforms traditional ML training for a given structure of the classifier, reducing error rates up to 18%. It is interesting to note that improving performance of HMM-HMT models in sequential pattern recognition tasks is important because no engineered feature extraction stage is required in such classifiers. Those feature extraction stages are often heuristic and very specific for the application. In this regard, pattern recognizers based on HMM-HMT models would be essentially similar for a broad range of applications.

Fully untied models were used in these developments and the specific structure of the HMM-HMT models was assumed known. While this structure can be chosen, for instance, using k-fold cross validation, having better alternatives for selecting it automatically would be useful in practice. When the availability of training data is too limited, tying parameters should also be useful to reduce the number of parameters to estimate. Nevertheless, choosing what parameters to tie should be carried out using rigurous tests that need to be developed for these models. Both points will be addressed in future work. It should be noted, nevertheless, that the same statements are valid for almost all types of HMM-based classifiers.

From an applications point of view, up to date the proposed algorithm for MCE training of HMM-HMT models has been used only with one-dimensional sequences. As the most important applications of HMT lies in imaging science, extensions of the proposed method to a bi-dimensional domain seems promising and will also be explored.

In the second part of the thesis, linear dimension reduction for GHMM-based classifiers was revisited, taking care of information loss that can be important to discriminate between classes. The framework of sufficient dimension reduction, which explicitely accounts for information retention, allowed us to analyze existing methods often used with GHMM-based classifiers and to propose new methods that achieve optimality in the sense of sufficiency. Both LDA and HLDA were analyzed in this framework and it was emphasized that the LAD estimator provides a natural way to deal with normal data, as it does not impose any restrictive constraint on the covariance structure of the populations.

On one hand, understanding LDA under the SDR methodology confirmed that LDA is optimal only when the Gaussian data have constant covariance matrix over all the classes. In addition, this analysis provided a ML interpretation for LDA that differs from the one that is commonly referred to in HMM-based applications. We have shown that such interpretation of LDA as a special case of HLDA assumes additional structure on the covariance matrices, not just being the same for all the populations.

Regarding HLDA, it was shown that this reduction method can always retain all the class information provided it projects the original features to a subspace that is large enough. Nevertheless, the needed directions may be significantly more than the minimum attainable, as achieved using LAD. This can be seen from another point of view. In applications, the dimension of the retained subspace is often fixed *a priori* because of practical considerations. Because HLDA is not optimal, it usually loses more relevant information than LAD for the fixed dimension. In addition, the HLDA estimator has no invariance property, which means that it changes completely under full-rank transformation of the features.

The lack of optimality of HLDA is due to the special covariance structure of the normal populations assumed by the method, which results from imposing strong independence in the transformed domain between the subset of discriminative directions and the rest of the coordinates that are equally distributed over all the classes. It turns out that this requirement of independence is not actually needed to reject those equally distributed coordinates as being relevant for classification. That flexibility is exploited by LAD to achieve a reduction that loses no information, is minimal, is equivariant and, unlike LDA and HLDA, imposes no constraints on the covariance of the models.

To the best of our knowledge, the LAD estimator had not been used previously in applications, neither yet in HMM-based classifiers. Using simulations, we strieved to emphasize the equivariance property of this estimator, which is important in applications and it is not a claimed attribute of other methods. Computational complexity for LAD is in the same order that for HLDA. Furthermore, it has been proved analytically that LAD performs well even when data deviates from normality. Summing up all these good properties, it seems clear that LAD is a better alternative to HLDA in GHMM-based applications. It was shown also that extending the method from normal populations to HMM is relatively easy. Though this extension follows the same guidelines as in HLDA, it should be clear that the resulting method has a theoretical background, is optimal in the sense of information retention and does not require a special structure on the covariance matrix of the observation models of the HMM.

Nevertheless, if the data were normally distributed satisfying the covariance structure assumed in HLDA but the minimal reduction were smaller than the one provided for that method, LAD would estimate the minimal reduction but lossing efficiency. To address this case, a new estimator LAD2 was introduced that both exploits the covariance structure of the data and achieves a minimal reduction.

On the computational side, the sufficiency approach led us to optimization algorithms with orthogonality matrix constraints. Though this is the standard practice in SDR, it was not in the implementations of LDA and HLDA used for instance in speech recognition. Understanding these methods under the sufficiency framework allowed us to implement them with the same tools used for SDR. These orthogonality-constrained implementations showed improved efficiency over the more standard unconstrained optimization.

Finally, it is important to emphasize that the methods and implementations discussed here estimate the columns of the projection matrix jointly, not in a sequential fashion. This is important to guarantee that the obtained estimate actually achieves the MLE.

Understanding HLDA under the sufficiency framework also allowed us to derive methods to infer about the dimension d of the reduced subspace that is sufficient to retain all the class information. We explored Akaike and Bayes information criteria, along with likelihood-ratio tests and permutation tests. Inference on d by BIC was found specially good, taking into account computational load.

Simulations were used to highlight the main points of all of these developments, and an example using a real dataset of handwritten digits confirmed the advantages of using LAD over HLDA. In this example, projecting the features from a 16-dimensional space to a subspace of dimension 2 and classifying in this smaller subspace, error rate was 5% for HLDA and 2% for LAD, which implies an improvement of 60% using the latter.

Future work should address extensive experiments to quantify the performance of LAD and LAD2 in HMM-based classifiers targeted to real-life applications, in order to verify if their theoretical advantages translate into practical interest. In this regard, extensions of the methodology to allow for multiple subspace projections is also of importance. Furthermore, in current developments of the SDR methodology, all the original features are linearly combined and then just a few of those linear combinations are retained. In future work, it would be interesting to explore adding variable selection procedures to reject some coordinates from the linear combinations. In addition, nonlinear sufficient dimension reduction is a field hardly addressed yet that can be explored.

APPENDIX A

Proofs for Section 3.3.3

A.1 Updating formulas for observation models

Let us consider the training formulas for the Gaussian means. We begin noting that the discriminant functions read:

$$\begin{aligned}
g_j(\mathbf{W};\Theta) &= \left|\log\left(\max_{\mathbf{q},\mathbf{R}}\left\{\mathcal{L}_{\vartheta_j}(\mathbf{W},\mathbf{q},\mathbf{R})\right\}\right)\right| \\
&= -\log\left(\max_{\mathbf{q},\mathbf{R}}\left\{\prod_{t=1}^{T} a_{q^{t-1}q^t}\prod_{\forall u}\epsilon^{q^t}_{u,r^t_u r^t_{\rho(u)}} f^{q^t}_{u,r^t_u}(w^t_u)\right\}\right) \\
&= -\sum_t \log a_{\bar{q}^{t-1}\bar{q}^t} - \sum_t\sum_{\forall u}\log \epsilon^{\bar{q}^t}_{u,\bar{r}^t_u\bar{r}^t_{\rho(u)}} - \sum_t\sum_{\forall u}\log f^{\bar{q}^t}_{u,\bar{r}^t_u}(w^t_u) \ ,
\end{aligned}$$

where, $\bar{q}^t$ and $\bar{r}^t$ refer to states in the external HMM and the corresponding HMT model, respectively, that achieve the maximum joint likelihood. To find (3.16), we know that we need

$$\begin{aligned}
\frac{\partial \ell_i(\mathbf{W};\Theta)}{\partial \tilde{\mu}^{(j)k}_{u,m}} &= \frac{\mathrm{d}\ell_i(\mathbf{W};\Theta)}{\mathrm{d}d_i(\mathbf{W};\Theta)}\frac{\partial d_i(\mathbf{W};\Theta)}{\partial g_i(\mathbf{W};\Theta)}\frac{\partial g_i(\mathbf{W};\Theta)}{\partial \tilde{\mu}^{(j)k}_{u,m}} \\
&= -\zeta\frac{\partial g_i(\mathbf{W};\Theta)}{\partial \tilde{\mu}^{(j)k}_{u,m}} \\
&= -\zeta\frac{\partial \sum_t \sum_{\forall u}\log f^{\bar{q}^t}_{u,\bar{r}^t_u}(w^t_u)}{\partial \tilde{\mu}^{(j)k}_{u,m}} \ .
\end{aligned}$$

In the expression above, we used ζ defined in Section 3.3.3. As observations in a node depends only on the state of that node, we have

$$\frac{\partial \ell_i(\mathbf{W};\Theta)}{\partial \tilde{\mu}^{(j)k}_{u,m}} = -\zeta \frac{\partial \sum_t \log f^{\bar{q}^t}_{u,\bar{r}^t_u}(w^t_u)}{\partial \tilde{\mu}^{(j)k}_{u,m}}.$$

As the sum takes into account only the most likely states in the node of the HMT related to the most likely state of the HMM in a given frame, we write

$$\begin{aligned}\frac{\partial \ell_i(\mathbf{W};\Theta)}{\partial \tilde{\mu}^{(j)k}_{u,m}} &= -\zeta \sum_t \delta(\bar{q}^t - k, \bar{r}^t_u - m) \frac{\partial \log f^{\bar{q}^t}_{u,\bar{r}^t_u}(w^t_u)}{\partial \tilde{\mu}^{(j)k}_{u,m}}. \\ &= -\zeta \sum_t \delta(\bar{q}^t - k, \bar{r}^t_u - m) \frac{\partial \mu^{(j)k}_{u,m}}{\partial \tilde{\mu}^{(j)k}_{u,m}} \frac{\partial \log f^{\bar{q}^t}_{u,\bar{r}^t_u}(w^t_u)}{\partial \mu^{(j)k}_{u,m}}.\end{aligned}$$

Noting that $\partial \mu^{(j)k}_{u,m} / \partial \tilde{\mu}^{(j)k}_{u,m} = \sigma^{(j)k}_{u,m}$ and that we are using an univariate Gaussian distribution for $f^{\bar{q}^t}_{u,\bar{r}^t_u}(w^t_u)$, we get (3.16).

The steps to derive the updating formulas for the Gaussian variances are completely analogous.

A.2 Updating formulas for transition probabilities

The procedure applied above also works well for transition probabilities, both in each HMT and in the external HMM of the whole HMM-HMT. Let us consider the estimation of the transition probabilities in the internal HMT. Reasoning as above, we just need

$$\frac{\partial \ell_i(\mathbf{W};\Theta)}{\partial \tilde{\epsilon}^{(i)k}_{u,mn}} = -\zeta \frac{\sum_t \log \epsilon^{\bar{q}^t}_{u,\bar{r}^t_u \bar{r}^t_{\rho(u)}}}{\partial \tilde{\epsilon}^{(i)k}_{u,mn}}.$$

Remembering of the transformation for this transition probabilities and proceeding as before to account for the most likely states in each frame, we get

$$\begin{aligned}\frac{\partial \ell_i(\mathbf{W};\Theta)}{\partial \tilde{\epsilon}^{(i)k}_{u,mn}} &= -\zeta \sum_t \sum_p \frac{\partial \epsilon^{(i)k}_{u,pn}}{\partial \tilde{\epsilon}^{(i)k}_{u,mn}} \frac{\partial \log \epsilon^{\bar{q}^t}_{u,\bar{r}^t_u \bar{r}^t_{\rho(u)}}}{\partial \epsilon^{(i)k}_{u,pn}} \\ &= -\zeta \sum_t \sum_p \delta(\bar{q}^t - k, \bar{r}^t_u - p, \bar{r}^t_{\rho(u)} - n) \frac{\partial \epsilon^{(i)k}_{u,pn}}{\partial \tilde{\epsilon}^{(i)k}_{u,mn}} \frac{\partial \log \epsilon^{k}_{u,pn}}{\partial \epsilon^{(i)k}_{u,pn}}.\end{aligned}$$

We now see that for $p \neq m$, we have $\partial\epsilon^{(i)k}_{u,pn}/\partial\tilde{\epsilon}^{(i)k}_{u,mn} = -\epsilon^{(i)k}_{u,pn}\epsilon^{(i)k}_{u,mn}$ and for $p = m$ we have $\partial\epsilon^{(i)k}_{u,pn}/\partial\tilde{\epsilon}^{(i)k}_{u,mn} = \epsilon^{(i)k}_{u,mn}(1 - \epsilon^{(i)k}_{u,mn})$. Replacing these results in the formula for the gradient and reordering, we get (3.18). An analogous procedure applies to derive the updating formulas for transition probabilities in the external HMM.

APPENDIX B

Proofs for Section 4.4.3

Let $\mathbf{X}|Y \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Delta}_y)$, with

$$\begin{aligned}\boldsymbol{\mu}_y &= \boldsymbol{\mu} + \boldsymbol{\rho}\boldsymbol{\Omega}\mathbf{A}\boldsymbol{\nu}_y \ , \\ \boldsymbol{\Delta}_y &= \boldsymbol{\rho}\boldsymbol{\Omega}\boldsymbol{\rho}^T + \boldsymbol{\rho}_0\boldsymbol{\Omega}_0\boldsymbol{\rho}_0^T + \boldsymbol{\rho}\boldsymbol{\Omega}\mathbf{A}\mathbf{T}_y\mathbf{A}^T\boldsymbol{\Omega}\boldsymbol{\rho}^T \ ,\end{aligned}$$

so that the central subspace is $\boldsymbol{\alpha} = \boldsymbol{\rho}\mathbf{A}$. Estimation of the parameters in model (B.1) is facilitated by centering so that the MLE of $\boldsymbol{\mu}$ is $\bar{\mathbf{X}}$. The transformed vectors $\boldsymbol{\rho}^T\mathbf{X}_y$ and $\boldsymbol{\rho}_0^T\mathbf{X}_y$ are independent, with means $\boldsymbol{\rho}^T\boldsymbol{\mu} + \boldsymbol{\Omega}\mathbf{A}\boldsymbol{\nu}_y$ and $\boldsymbol{\rho}^T\boldsymbol{\mu}$, and covariance matrices $\boldsymbol{\Omega} + \boldsymbol{\Omega}\mathbf{A}\mathbf{T}_y\mathbf{A}^T\boldsymbol{\Omega}$ and $\boldsymbol{\Omega}_0$, respectivly. Thus the likelihood factors in these quantities, and leads to the log-likelihood maximized over all the parameters

$$\mathcal{L}(\boldsymbol{\rho}, \mathbf{A}, \boldsymbol{\Omega}_0, \boldsymbol{\Omega}|d, u) = \mathcal{L}_0 + \mathcal{L}_1(\boldsymbol{\rho}_0, \boldsymbol{\Omega}_0|u) + \mathcal{L}_2(\boldsymbol{\rho}, \mathbf{A}, \boldsymbol{\Omega}|d, u)$$

where

$$\begin{aligned}
\mathcal{L}_0 &= -(np/2)\log(2\pi) \\
\mathcal{L}_1(\boldsymbol{\rho}_0, \boldsymbol{\Omega}_0|u) &= -(n/2)\log|\boldsymbol{\Omega}_0| - \\
&\quad -\frac{1}{2}\sum_{y=1}^{H}\sum_{i=1}^{n_y}\{\boldsymbol{\rho}_0^T(\mathbf{X}_{yi}-\bar{\mathbf{X}})\}^T\boldsymbol{\Omega}_0^{-1}\boldsymbol{\rho}_0^T(\mathbf{X}_{yi}-\bar{\mathbf{X}}) \\
\mathcal{L}_2(\boldsymbol{\rho}, \mathbf{A}, \boldsymbol{\Omega}|d,u) &= -\frac{n}{2}\log|\boldsymbol{\Omega}+\boldsymbol{\Omega}\mathbf{A}\mathbf{T}_y\mathbf{A}^T\boldsymbol{\Omega}| - \\
&\quad -\frac{1}{2}\sum_{y=1}^{H}\sum_{i=1}^{n_y}\mathbf{C}^T(\boldsymbol{\Omega}+\boldsymbol{\Omega}\mathbf{A}\mathbf{T}_y\mathbf{A}^T\boldsymbol{\Omega})^{-1}\mathbf{C}.
\end{aligned}$$

Here we have used $\mathbf{C} = \boldsymbol{\rho}^T(\mathbf{X}_{yi}-\bar{\mathbf{X}}) - \boldsymbol{\Omega}\mathbf{A}\boldsymbol{\nu}_y$. It follows that $\mathcal{L}_1$ is maximized over $\boldsymbol{\Omega}_0$ by $\widehat{\boldsymbol{\Omega}}_0 = \boldsymbol{\rho}_0^T\tilde{\boldsymbol{\Sigma}}\boldsymbol{\rho}_0$. Substituting back, we find the following partially maximized form for $\mathcal{L}_1$:

$$\mathcal{L}_1(\boldsymbol{\rho}_0|u) = -(n/2)\log|\boldsymbol{\rho}_0^T\tilde{\boldsymbol{\Sigma}}\boldsymbol{\rho}_0| - n(p-u)/2.$$

For fixed $\boldsymbol{\rho}$, the log likelihood summand $\mathcal{L}_2$ is in the same form as the likelihood considered for LAD model, with the parameters and variables redefined as $\boldsymbol{\Delta} \to \boldsymbol{\Omega}$, $p \to u$, $\boldsymbol{\alpha} \to \mathbf{A}$ and $(\mathbf{X}_y - \bar{\mathbf{X}}) \to \boldsymbol{\rho}^T(\mathbf{X}_y - \bar{\mathbf{X}})$. Thus for fixed $\boldsymbol{\rho}$ we have from (4.6) a partially maximized version of $\mathcal{L}_2$:

$$\begin{aligned}
\mathcal{L}_2(\boldsymbol{\rho}|d,u) &= -un/2 + n/2\log|\mathbf{A}^T\boldsymbol{\rho}^T\tilde{\boldsymbol{\Sigma}}\boldsymbol{\rho}\mathbf{A}| \\
&\quad -n/2\log|\boldsymbol{\rho}^T\tilde{\boldsymbol{\Sigma}}\boldsymbol{\rho}| - 1/2\sum_{y=1}^{H} n_y\log|\mathbf{A}^T\boldsymbol{\rho}^T\tilde{\boldsymbol{\Sigma}}_y\boldsymbol{\rho}\mathbf{A}|.
\end{aligned}$$

Substituting back in $\mathcal{L}$ we get

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\rho}|d,u) &= -(pn/2)[1+\log(2\pi)] + n/2\log|\mathbf{A}^T\boldsymbol{\rho}^T\tilde{\boldsymbol{\Sigma}}\boldsymbol{\rho}\mathbf{A}| - \\
&\quad -n/2\log|\boldsymbol{\rho}^T\tilde{\boldsymbol{\Sigma}}\boldsymbol{\rho}| - 1/2\sum_{y=1}^{H} n_y\log|\mathbf{A}^T\boldsymbol{\rho}^T\tilde{\boldsymbol{\Sigma}}_y\boldsymbol{\rho}\mathbf{A}| - \\
&\quad -n/2\log|\boldsymbol{\rho}^T\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{\rho}|.
\end{aligned}$$

Bibliography

[1] M. Afify, X. Li, and H. Jiang, "Statistical analysis of minimum classification error learning for gaussian and hidden Markov model classifiers," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 2405–2417, 2007.

[2] L. Bahl, P. Brown, P. D. Souza, and R. Mercer, "Maximum mutual information estimation of HMM parameters for speech recognition," in *Proc. of the Int. Conf. on Audio, Speech, and Signal processing (ICASSP86)*, 1986, pp. 49–52.

[3] P. Baldi and S. Brunak, *Bioinformatics: The Machine Learning Approach.* Cambridge, Massachusetts: MIT Press, 2001.

[4] L. Baum, T. Petric, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Annals Mathematical Statistics*, vol. 41, pp. 164–171, 1970.

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

[6] C. Bishop, *Pattern Recognition and Machine Learning.* New York: Springer, 2007.

[7] J. C. Bremer, R. R. Coifman, M. Maggioni, and A. D. Szlam, "Diffusion wavelet packets," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 95 – 112, 2006, diffusion Maps and Wavelets.

[8] K. Burnham and D. Anderson, *Model Selection and Multimodel Inference.* New York: Wiley, 2002.

[9] J. Cai and Z.-Q. Liu, "Hidden markov models with spectral features for 2d shape recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 12, pp. 1454–1458, Dec. 2001.

[10] N. Campbell, "Canonical variate analysis - a general model formulation," *Australian Journal of Statistics*, vol. 26, pp. 86–96, 1984.

[11] O. Cappé, E. Moulines, and T. Ryden, *Inference in Hidden Markov Models.* New York: Springer, 2005.

[12] Y. Chikuse, *Statistics on Special Manifolds.* New York: Springer, 2003.

[13] W. Chou, "Minimum classification error rate (MCE) approach in pattern recognition," in *Pattern Recognition in Speech and Language Processing*, W. Chou and B. Juang, Eds. CRC Press, 2003, pp. 1–49.

[14] W. Chou, B.-H. Huang, and C.-H. Lee, "Segmental GPD training for HMM based speech recognition," in *Proc. of the Int. Conf. on Audio, Speech, and Signal processing (ICASSP92)*, vol. 1, 1992, pp. 473–476.

[15] R. R. Coifman and M. Maggioni, "Diffusion wavelets," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 53 – 94, 2006, diffusion Maps and Wavelets.

[16] R. Cook, "Using dimension reduction subspaces to identify important inputs in models of physical systems," 1994, pp. 18–25.

[17] ——, *Regression Graphics.* New York: Wiley, 1998.

[18] ——, "Fisher lecture: Dimension reduction in regression (with discussion)," *Statistical Science*, vol. 22, pp. 1–26, 2007.

[19] R. Cook and L. Forzani, "Letters to the editor: Response to zhu and hastie," *Journal of the American Statistical Association, to appear.*

[20] ——, “Likelihood-Based sufficient dimension reduction,” *Journal of the American Statistical Association*, vol. 104, no. 485, pp. 197–208, 2008.

[21] ——, “Principal fitted components in regression,” *Statistical Science*, vol. 23, pp. 485–501, 2008.

[22] R. Cook, L. Forzani, and D. Tomassi, “LDR: a package for likelihood-based sufficient dimension reduction,” *Journal of Statistical Software. Accepted.*

[23] R. Cook and X. Yin, “Dimension reduction and visualization in discriminant analysis (with discussion),” *Australia New Zeland Journal of Statistics*, pp. 18–25, 1994.

[24] M. Crouse, R. Nowak, and R. Baraniuk, “Wavelet-based statistical signal processing using hidden Markov models,” *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 886–902, 1998.

[25] A. Dainotti, W. de Donato, A. Pescape, and P. Salvo Rossi, “Classification of network traffic via packet-level hidden markov models,” in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, 30 2008.

[26] K. Das and Z. Nenadic, “Approximate information discriminant analysis: A computationally simple heteroscedastic feature extraction technique,” *Pattern Recognition*, vol. 41, no. 5, pp. 1565–1574, 2008.

[27] I. Daubechies, *Ten Lectures on Wavelets.* Society for Industrial and Applied Mathematics, 1992.

[28] J. R. Deller, J. G. Proakis, and J. H. Hansen, *Discrete-Time Processing of Speech Signals.* NewYork: Macmillan Publishing, 1993.

[29] A. Dempster, N. Laird, and D. Durbin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society, Series B*, vol. 39, pp. 1–38, 1977.

[30] ——, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society, Series B*, vol. 39, pp. 1–38, 1977.

[31] D. Donoho and I. Johnstone, “Adapting to unknown smoothness by wavelet shrinkage,” *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1200–1224, 1995.

[32] M. Duarte, M. Wakin, and R. Baraniuk, “Wavelet-domain compressive signal reconstruction using a hidden Markov tree model,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*, 31 2008-April 4 2008, pp. 5137–5140.

[33] R. Duda, P. Hart, and D. Stork, *Pattern Classification, Second Edition.* Wiley, 2000.

[34] J.-B. Durand, P. Gonçalvès, and Y. Guédon, “Computational methods for hidden Markov trees,” *IEEE Transactions on Signal Processing*, vol. 52, no. 9, pp. 2551–2560, 2004.

[35] M. Eaton, *Multivariate Statistics.* New York: Wiley, 1983.

[36] A. El-Yacoubi, M. Gilloux, R. Sabourin, and C. Suen, “An hmm-based approach for off-line unconstrained handwritten word modeling and recognition,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 8, pp. 752–760, Aug. 1999.

[37] R. Elliott, L. Aggoun, and J. B. Moore, *Hidden Markov Models: Estimation and Control.* New York: Springer, 1995.

[38] R. Ferrari, H. Zhang, and C. Kube, “Real-time detection of steam in video images,” *Pattern Recognition*, vol. 40, no. 3, pp. 1148 – 1159, 2007.

[39] G. Fink, *Markov Models for Patten Recognition: from Theory to Application.* New York: Springer, 2007.

[40] R. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics*, vol. 7, pp. 179–188, 1936.

[41] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* New York: Academic Press, 1990.

[42] M. Gales, “Maximum likelihood multiple subspace projections for hidden markov models,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, pp. 37–47, 2002.

[43] S. Graja and J.-M. Boucher, “Hidden Markov tree model applied to ECG delineation,” *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 6, pp. 2163–2168, Dec. 2005.

[44] X. He and L. Deng, “A new look at discriminative training for hidden Markov models,” *Pattern Recognition Letters*, vol. 28, pp. 1285–1294, 2007.

[45] X. He, L. Deng, and W. Chou, “Discriminative learning in sequential pattern recognition: a unifying review for optimization-based speech recognition,” *IEEE Signal Processing Magazine*, vol. 25, pp. 14–36, 2008.

[46] Z. He, X. You, and Y. Y. Tang, “Writer identification of chinese handwriting documents using hidden Markov tree model,” *Pattern Recognition*, vol. 41, no. 4, pp. 1295 – 1307, 2008.

[47] J. Hu, M. Brown, and W. Turin, “Hmm based online handwriting recognition,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 10, pp. 1039–1045, Oct. 1996.

[48] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: a Guide to Theory, Algorithm and System Development*. New Jersey: Prentice Hall, 2001.

[49] A. Izenman, *Modern Multivariate Statistical Techniques. Regression, Classification and Manifold Learning*. New York: Springer, 2008.

[50] A. Jain, R. Duin, and J. Mao, “Statistical pattern recognition: a review,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 4–37, 2000.

[51] F. Jelinek, *Statistical Methods for Speech Recognition*. Cambrige, Masachussets: MIT Press, 1999.

[52] H. Jiang, “Discriminative training of HMMs for automatic speech recognition: A survey,” *Computer, Speech and Language, in press*, 2009.

[53] I. Jolliffe, *Principal Component Analysis, Second Edition*. New York: Springer, 2002.

[54] 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, pp. 257–265, 1997.

[55] S. Katagiri, B.-H. Juang, and C. Lee, “Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method,” *Proceedings of the IEEE*, vol. 86, pp. 2345–2373, 1998.

[56] N. Kumar, “Investigation of silicon auditory models and generalization of linear discriminant analysis for improved speech recognition,” Ph.D. dissertation, John Hopkins University, Baltimore, 1997.

[57] N. Kumar and A. Andreou, “Heteroscedastic discriminant analysis and reduced-rank HMMs for improved speech recognition,” *Speech Communication*, vol. 26, pp. 283–297, 1998.

[58] S. Lefkimmiatis, G. Papandreou, and P. Maragos, “Photon-limited image denoising by inference on multiscale models,” in *Proc. IEEE Int. Conf. on Image Processing (ICIP-08)*, San Diego, CA, 2008, pp. 2332–2335.

[59] F. Li, X. Jia, and D. Fraser, “Universal HMT based super resolution for remote sensing images,” in *15th IEEE International Conference on Image Processing (ICIP 2008)*, Oct. 2008, pp. 333–336.

[60] K. Li, “Sliced inverse regression for dimension reduction (with discussion),” *Journal of the American Statistical Association*, vol. 86, pp. 316–342, 1991.

[61] R. Lippert and A. Edelman, “Nonlinear eigenvalue problems with orthogonality constraints,” in *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, Eds. SIAM, 2000.

[62] C.-S. Liu, C.-H. Lee, B.-H. Juang, and A. Rosenberg, “A study on minimum error discriminative training for speaker recognition,” *J. of the Acoustical Society of America*, vol. 97, no. 1, pp. 637–648, 1995.

[63] M. Loog, R. P. W. Duin, and R. Haeb-Umbach, “Multiclass linear dimension reduction by weighted pairwise fisher criteria,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 7, pp. 762–766, 2001.

[64] M. Loog and R. P. W. Duin, “Linear dimensionality reduction via a heteroscedastic extension of lda: The chernoff criterion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 732–739, 2004.

[65] S. Mahadevan, *Representation Discovery Using Harmonic Analysis.* Morgan & Calypool, 2008.

[66] J. E. B. Maia and R. Holanda Filho, "Internet traffic classification using a hidden markov model," in *Hybrid Intelligent Systems (HIS), 2010 10th International Conference on*, 2010, pp. 37 –42.

[67] S. Mallat, *A Wavelet Tour of signal Processing*, 2nd ed. Academic Press, 1999.

[68] R. Mamon and R. Elliott, *Hidden Markov Models in Finance.* New York: Springer, 2010.

[69] E. McDermott, T. Hazen, J. L. Roux, A. Nakamura, and S. Katagiri, "Discriminative training for large-vocabulary speech recognition using minimum classification error," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 203–223, 2007.

[70] E. McDermott and S. Katagiri, "A derivation of minimum classification error from the theoretical classification risk using Parzen estimation," *Computers, Speech and Language*, vol. 18, pp. 102–122, 2004.

[71] D. H. Milone and L. E. D. Persia, "An EM algorithm to learn sequences in the wavelet domain," *Lecture Notes in Computer Science*, vol. 4827, pp. 518–528, 2007.

[72] D. Milone, L. Di Persia, and D. Tomassi, "Signal denoising with hidden Markov models using hidden Markov trees as observation densities," in *Proc. of the IEEE Workshop on Machine Learning for Signal Processing*, Cancún, Mexico, aceptado para publicación 2008.

[73] D. H. Milone, L. E. D. Persia, and M. E. Torres, "Denoising and recognition using hidden Markov models with observation distributions modeled by hidden Markov trees," *Pattern Recognition, in press*, 2009.

[74] E. Mor and M. Aladjem, "Boundary refinements for wavelet-domain multiscale texture segmentation," *Image and Vision Computing*, vol. 23, no. 13, pp. 1150 – 1158, 2005.

[75] A. Nadas, "A decision theoretic formulation of a training problem in speech recognition and a comparison of training by unconditional versus conditional maximum likelihood," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, no. 4, pp. 814–817, 1983.

[76] A. Nait-Ali, *Advanced Biosignal Processing.* New York: Springer, 2009.

[77] Z. Nenadic, "Information discriminant analysis: Feature extraction with an information-theoretic objective," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1394–1407, 2007.

[78] G. Papandreou, P. Maragos, and A. Kokaram, "Image inpainting with a wavelet domain hidden Markov tree model," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP-2008)*, Las Vegas, Nevada, 2008, pp. 773–776.

[79] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Cambridge University, Cambridge, UK, 2004.

[80] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition.* New Jersey: Prentice-Hall, 1993.

[81] ——, *Fundamentals of Speech Recognition.* New Jersey: Prentice-Hall, 1993.

[82] V. R. Rallabandi and V. S. Rallabandi, "Rotation-invariant texture retrieval using wavelet-based hidden Markov trees," *Signal Processing*, vol. 88, no. 10, pp. 2593 – 2598, 2008.

[83] G. Saon, M. Padmanabhan, R. Gopinath, and S. Chen, "Maximum likelihood discriminant feature spaces," *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 2, pp. 1129–1132, 2000.

[84] C. Tantibundhit, J. Boston, C. Li, J. Durrant, S. Shaiman, K. Kovacyk, and A. El-Jaroudi, "New signal decomposition method based speech enhancement," *Signal Processing*, vol. 87, no. 11, pp. 2607 – 2628, 2007.

[85] Y. Tian, J. Wang, J. Zhang, and Y. Ma, "A contextual hidden Markov tree model image denoising using a new nonuniform quincunx directional filter banks," in *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP 2007)*, vol. 1, Nov. 2007, pp. 151–154.

[86] D. Tomassi, L. Forzani, D. Milone, and R. Cook, "Likelihood-based sufficient dimension reduction for statistical pattern recognition," *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[87] ——, "Sufficient dimension reduction for hidden Markov models," *In preparation*, 2010.

[88] D. Tomassi, D. Milone, and L. Forzani, "Minimum classification error training of hidden Markov models for sequential data in the wavelet domain," *Revista Iberoamericana de Inteligencia Artificial*, vol. 13, no. 44, pp. 46–55, 2009.

[89] ——, "Minimum classification error training for sequential data in the wavelet domain," *Pattern Recognition, en prensa*, 2010, doi:10.1016/j.patcog.2010.07.010.

[90] S. Veltman and R. Prasad, "Hidden markov models applied to on-line handwritten isolated character recognition," *Image Processing, IEEE Transactions on*, vol. 3, no. 3, pp. 314–318, May 1994.

[91] A. Willsky, "Multiresolution Markov models for signal and image processing," *Proc. of the IEEE*, vol. 90, no. 8, pp. 1396–1458, Aug. 2002.

[92] P. Woodland and D. Povey, "Large scale discriminative training of hidden Markov models for speech recognition," *Computer, Speech and Language*, vol. 16, pp. 25–47, 2002.

[93] C. Yen, S.-S. Kuo, and C.-H. Lee, "Minimum error rate training for PHMM-based text recognition," *IEEE Trans. on Image Proc.*, vol. 8, no. 8, pp. 1120–1124, 1999.

[94] J. Zhang and Y. Liu, "Svm decision boundary based discriminative subspace induction," *Pattern Recognition*, vol. 38, no. 10, pp. 1746–1758, 2005.

[95] Y. Zhang, Y. Zhang, Z. He, and X. Tang, "Multiscale fusion of wavelet-domain hidden Markov tree through graph cut," *Image and Vision Computing*, vol. In Press, Corrected Proof, pp. –, 2009.

[96] H. Zhou, D. Karakos, S. Khudanpur, A. Andreou, and C. Priebe, "On projections of gaussian distributions using maximum likelihood criteria," 2009, pp. 431–438.

[97] M. Zhu and T. J. Hastie, "Feature extraction for non-parametric discriminant analysis," *Journal of Computational and Graphical Statistics*, pp. 101–120, 2003.

[98] ——, “Letter to the editor about the article by cook and forzani, likelihood-based sufficient dimension reduction,” *Journal of the American Statistical Association*, vol. 105, p. 880, 2010.

[99] V. Zue, S. Sneff, and J. Glass, “Speech database development: TIMIT and beyond.” *Speech Communication*, vol. 9, pp. 351–356, 1990.