etsit
escuela técnica superior de
ingeniería de telecomunicación

INOC UPCT
ESCUELA INTERNACIONAL DE DOCTORADO

TESIS 2019

APRENDIZAJE MÁQUINA BASADO EN SITUACIONES DE TANGENCIA DISJUNTA

Juan Antonio Martínez García

Universidad Politécnica de Cartagena

Universidad Politécnica de Cartagena | Campus de Excelencia Internacional

APRENDIZAJE MÁQUINA BASADO EN SITUACIONES DE TANGENCIA DISJUNTA

Tecnologías de la Información y Comunicaciones

Autor: Juan Antonio Martínez García

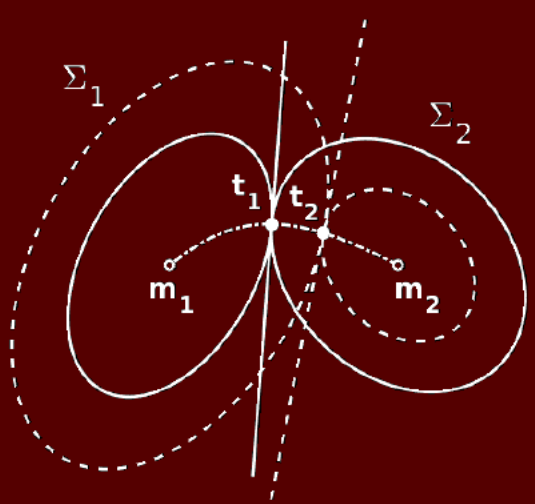
Cartagena (2019)

Juan Antonio Martínez García

Aprendizaje Máquina basado en Situaciones de Tangencia Disjunta

TESIS DOCTORAL

Programa de Tecnologías de la Información y Comunicaciones



upct

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

DEPARTAMENTO DE TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES

Aprendizaje Máquina basado en Situaciones de Tangencia Disjunta

TESIS DOCTORAL

Programa de Tecnologías de la Información y Comunicaciones

Autor: D. Juan Antonio Martínez García
juan.antonio.mtnez@gmail.com

Director: Dr. D. José Luis Sancho Gómez
josel.sancho@upct.es

Cartagena, 2019

Universidad Politécnica de Cartagena

DT-16

CONFORMIDAD DE SOLICITUD DE AUTORIZACIÓN DE DEPÓSITO DE TESIS DOCTORAL POR EL DIRECTOR DE LA TESIS

D. José Luis Sancho Gómez, Director de la Tesis doctoral "Aprendizaje Máquina basado en Situaciones de Tangencia Disjunta",

INFORMA:

Que la referida Tesis Doctoral ha sido realizada por D. Juan Antonio Martínez García, dentro del Programa de Doctorado en Tecnologías de la Información y Comunicaciones, dando mi conformidad para que sea presentada ante el Comité de Dirección de la Escuela Internacional de Doctorado para ser autorizado su depósito.

- Informe positivo sobre el plan de investigación y documento de actividades del doctorando emitido por el Director (**RAPI**).

La rama de conocimiento en la que esta tesis ha sido desarrollada es:

- Ingeniería y Arquitectura.

En Cartagena, a 18 de julio de 2019

EL DIRECTOR DE LA TESIS

[redacted]

Fdo: D. José Luis Sancho Gómez

COMITÉ DE DIRECCIÓN DE LA ESCUELA INTERNACIONAL DE DOCTORADO

Universidad Politécnica de Cartagena

DT-17

CONFORMIDAD DE DEPÓSITO DE TESIS DOCTORAL POR LA COMISIÓN ACADÉMICA DEL PROGRAMA

D. Jorge Larrey Ruiz, Presidente de la Comisión Académica del Programa de Doctorado en Tecnologías de la Información y Comunicaciones,

INFORMA:

Que la Tesis Doctoral titulada "Aprendizaje Máquina basado en Situaciones de Tangencia Disjunta", ha sido realizada, dentro del mencionado Programa de Doctorado, por D. Juan Antonio Martínez García, bajo la dirección y supervisión del Dr. D. José Luis Sancho Gómez.

En reunión de la Comisión Académica, visto que en la misma se acreditan los indicios de calidad correspondientes y la autorización del Director de la misma, se acordó dar la conformidad, con la finalidad de que sea autorizado su depósito por el Comité de Dirección de la Escuela Internacional de Doctorado.

- Evaluación positiva del plan de investigación y documento de actividades por el Presidente de la Comisión Académica del programa (**RAPI**).

La rama de conocimiento en la que esta tesis ha sido desarrollada es:

- Ingeniería y Arquitectura

En Cartagena, a 18 de julio de 2019

EL PRESIDENTE DE LA COMISIÓN ACADÉMICA

Fdo: D. Jorge Larrey Ruiz

El contenido de esta tesis esta duplicado en español e inglés

The content of this thesis is duplicated in Spanish and English.

English version is from page 100

Prólogo

¿Por qué estudiar un doctorado?

La razón práctica es validar el inicio de una carrera profesional relacionada con la investigación, en muchos casos dentro de la enseñanza, o tal vez en una del creciente número de empresas que requieren un doctorado para realizar i+D. Supone también una gran oportunidad de aprendizaje, una especialización práctica en algún campo en el que además se hace alguna aportación novedosa. Sin embargo, unos de los principales atractivos es que tal aprendizaje se realiza en unas situaciones distintas a las de una clase o un trabajo, en los que hay unas directrices fijas acerca de el contenido o un criterio de selección en base a la rentabilidad. El alumno de doctorado goza de una libertad que le permite explorar y tomar decisiones sobre la dirección de su trabajo. Se trata de entorno perfecto para quien disfruta aprendiendo, siendo guiado por un experto en la materia, el director de la tesis, del que también se aprende su forma de aprender.

"Dimidium facti, qui coepit, habet; sapere aude, incipe. [Tiene medio camino hecho quien ha empezado: atrévete a saber, ¡empieza!]"

— Horacio. Epist. I, 2

Aprendizaje automático (Machine learning)

El asunto elegido para esta tesis, el aprendizaje automático, es una rama de la inteligencia artificial que tiene como objetivo que las computadoras aprendan sin que explícitamente se les indique cómo hacerlo, sino a partir de la experiencia, a través de los datos suministrados. Esta inteligencia artificial forma parte principal de la denominada Cuarta Revolución Industrial en la que las fronteras existentes entre la física, lo digital y la biología se difuminan. El alcance real que tendrá en la transformación de la sociedad es inimaginable. Las actividades que realizarán las máquinas, algunas de ellas efectuadas en la actualidad por nosotros, después del temor inicial relativo a la pérdida de puestos de trabajo, plantearán la cuestión de qué capacidades son intrínsecamente humanas, una vez que se constate que aptitudes que pensábamos que eran propiedad de los humanos dejen de serlo.

"Somos el universo contemplándose a sí mismo."

— Carl Sagan

Agradecimientos

En primer lugar, quiero agradecer todo lo que soy a mis padres. Ellos trabajaron muy duro para que yo tuviera todas las oportunidades que me han permitido desarrollarme como persona en todas su facetas.

Lo valoro aún más si cabe, ya que ellos partieron de unas condiciones más adversas. Mi único mérito ha sido aprovechar algunas de las oportunidades que he tenido. De mi padre, ingeniero también, aprendí el amor por el conocimiento y la ciencia, la honestidad y bondad; todavía recuerdo cómo de niño prefería para quedarme dormido que me explicase el funcionamiento de las cosas en lugar de contarme cuentos. El murió durante la realización del proyecto final de carrera pero estoy seguro de que en vida ya concibió lo que está ocurriendo ahora y lo que está por venir. De mi madre, con formación artística, he aprendido y sigo aprendiendo todo lo que no puede ser medido y que constituye el eje principal de la vida, el amor, el esfuerzo incansable, la autoexigencia, la sensibilidad estética y la creatividad.

Quiero agradecer también al Dr. D. José Luis Sancho Gómez, Profesor Titular de la Universidad Politécnica de Cartagena e Investigador Principal del grupo de i+D de Tratamiento de Datos y Aprendizaje Máquina (TDAM), director de la tesis, la oportunidad de haber realizado el doctorado a partir del estudio de las situaciones de tangencia disjunta que el había iniciado años atrás. Quiero destacar su generosidad tanto en la transmisión de conocimientos como en su dedicación y consejos prácticos.

Quiero dedicar una mención especial al Dr. D. Aníbal R. Figueiras Vidal, Catedrático de Universidad en la Universidad Carlos III de Madrid e investigador de prestigio internacional, por la generosidad de su ayuda en la realización de los artículos científicos que han servido para dotar de contenido a esta tesis.

Recuerdo también a mis profesores del colegio San Pablo C.E.U. de Murcia, por enseñarme el valor del conocimiento, el trabajo riguroso y el juicio crítico.

Resumen

El objetivo de un clasificador es decidir, con el menor error posible, a qué clase pertenece una muestra o patrón. En esta tesis, se presenta una nueva interpretación de los discriminantes lineales, en la que son descritos en términos de situaciones de tangencia disjunta (*Disjoint Tangent Configurations*, DTC) establecidas entre las superficies elipsoidales de nivel de probabilidad resultantes de la caracterización de las distribuciones de las clases de los datos por sus dos primeros momentos, las medias y las matrices de covarianza. Éste es un marco común que permite el diseño y análisis de distintos discriminantes conocidos a través de una correspondencia analítica con otros métodos: el método paramétrico, que consiste en la minimización de una función de error en un espacio proyectado unidimensional para determinar los parámetros de la expresión matemática del discriminante, *e.g.*, el discriminante lineal de Fisher, el basado en matrices Scatter o el de Bayes, cuya expresión explícita es aún desconocida; y el método de optimización convexa minimax, que consiste en acotar y minimizar la probabilidad de clasificación errónea, *e.g.*, la solución del Hiperplano de Decisión Probabilística Minimax (*Minimax Probabilistic Decision Hyperplane*, MPDH) proporcionada por la Máquina de Probabilidad Minimax (*Minimax Probability Machine*, MPM), que minimiza el peor caso o máximo riesgo sobre todas las distribuciones posibles caracterizadas por los mismos primeros dos momentos, lo que es adecuado cuando las distribuciones de las clases de los datos son desconocidas o no reflejan las probabilidades a priori reales. También permite el diseño de nuevos discriminantes, como un discriminante de Fisher completo con un término independiente o el Quasi-Bayes, que es una aproximación geométrica del Bayes óptimo con una precisión similar y menor coste computacional, una ventaja general de DTC ya que es un método no iterativo.

Clasificación

DTC

Correspondencia analítica

Método paramétrico

Método de optimización convexa minimax

Minimax

Nuevos discriminantes

Bajo coste computacional

En la segunda parte de la tesis, las versiones no lineales de los discriminantes lineales DTC se construyen usando Redes de Funciones de Base Radial (*Radial Basis Function Networks*, RBFNs) con núcleos Gaussianos pre-entrenados mediante técnicas de cuantificación vectorial. De esta manera, se transforma el espacio de datos de entrada en un espacio superior con mayor separabilidad lineal en el que se resuelve el problema de clasificación con un discriminante DTC lineal. El discriminante DTC no lineal resultante mantiene las propiedades del discriminante DTC lineal original y permite resolver problemas más complejos con clases que no son linealmente separables.

DTC no lineales

RBFNs

Núcleos Gauss. pre-entrenados

Los experimentos muestran que los DTCs obtienen buenos resultados de precisión con un coste computacional competitivo en términos de tiempo de entrenamiento, debido a que es una solución no iterativa y a la ausencia de parámetros de entrenamiento que necesiten ser ajustados, y de requisitos de memoria en las versiones no lineales, en comparación con las redes de núcleos entrenadas globalmente.

DTC: buena precisión y coste computacional competitivo

Publicaciones

- Martínez-García, J.-A. y Sancho-Gómez, J.-L. (2018) Performance analysis of No-Propagation and ELM algorithms in classification. *Neural Comput. Appl.* doi: 10.1007/s00521-018-3353-0. http://rdcu.be/E68l
- Sancho-Gómez, J.-L., Martínez-García, J.-A., Ahalt, S. C. y Figueiras-Vidal, A. R. (2018) Linear discriminants described by disjoint tangent configurations. *Neurocomputing*, 316:345–356. doi: 10.1016/j.neucom.2018.08.010
- Sánchez-Morales, A., Sancho-Gómez, J.-L., Martínez-García, J.-A. y Figueiras-Vidal, A. R. (2019) Improving deep learning performance with missing values via deletion and compensation. *Neural Comput. Appl.* doi: 10.1007/s00521-019-04013-2
- Martínez-García, J.-A., Sancho-Gómez, J.-L., Sánchez-Morales, A. y Figueiras-Vidal, A. R. (2019) Designing non-linear minimax and related discriminants by disjoint tangent configurations applied to RBF networks. *Neurocomputing*. Unpublished. In revision

NEUROCOMPUTING

Impact Factor (2018): 4.072

JCR© Category	Rank in Category	Quartile in Category
Computer Science, Artificial Intelligence	**28 of 133**	**Q1**

Publisher: ELSEVIER SCIENCE BV,
PO BOX 211, 1000 AE Amsterdam, Netherlands
ISSN: 0925-2312 **eISSN:** 1872-8286
Research Domain: Computer Science

NEURAL COMPUTING & APPLICATIONS

Impact Factor (2018): 4.664

JCR© Category	Rank in Category	Quartile in Category
Computer Science, Artificial Intelligence	**21 of 133**	**Q1**

Publisher: SPRINGER LONDON LTD,
236 Grays Inn RD, 6th Floor, London WC1X 8HL, England
ISSN: 0941-0643 **eISSN:** 1433-3058
Research Domain: Computer Science

Índice

Nomenclatura

La siguiente lista describe los símbolos, siglas y abreviaturas usadas en el documento. Los vectores columna están en negrita y minúsculas, las matrices en mayúsculas y los escalares en minúsculas, estos dos últimos en texto sin formato.

Algoritmos

CV Validación Cruzada (*Cross-Validation*)

DTC Situaciones de Tangencia Disjunta (*Disjoint Tangent Configurations*)

FSCL Aprendizaje Competitivo Sensible a la Frecuencia (*Frequency Sensitive Competitive Learning*)

MEMPM Máquina de Probabilidad Minimax de Mínimo Error (*Minimum Error Minimax Probability Machine*)

MLP Perceptrón Multicapa (*Multilayer Perceptron*)

MPM Máquina de Probabilidad Minimax (*Minimax Probability Machine*)

SOCP Programa de Cono de Segundo Orden (*Second-Order Cone Program*)

SVM Máquina de Vectores Soporte (*Support Vector Machine*)

TM Método Teórico (*Theoretical Method*)

Iteradores

i Muestra i-ésima, i=$1, 2, \ldots, N$

j Clase j-ésima o nodo de salida j-ésimo, j=$1, 2, \ldots, \tilde{n}$, (excepto en clasificación binaria, j=$1, 2$, donde sólo hay un nodo de salida, $\tilde{n} = 1$)

k Nodo oculto k-ésimo, k=$1, 2, \ldots, m$

Datos de entrada (muestras)

$\mathbf{x}$ Datos de entrada, muestra (variable aleatoria), $\mathbf{x} = [x_1, x_2, \ldots, x_n]$

$\mathbf{x}^{(i)}$ Muestra i-ésima, $\mathbf{x}^{(i)} = [x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}]$

N Número de muestras

n Dimensión o número de variables o características de las muestras, número de nodos de entrada

C_j Clase j-ésima

P_j Probabilidad a priori de la clase j-ésima

$\mathbf{m}_j$ Vector media de las muestras de la clase j-ésima

Σ_j Matriz de covarianza de las muestras de la clase j-ésima

Discriminantes lineales

$\mathbf{w}$	Vector de pesos
$\mathbf{x}_0$	Vector de sesgos
ω_0	Valor umbral
$\mathcal{H}_{\mathrm{L}}$	Hiperplano, frontera de decisión lineal
h	Función de proyección de las muestras en la dirección $\mathbf{w}$ o espacio proyectado unidimensional
μ	Media de las muestras proyectadas en el espacio h
σ^2	Varianza de las muestras proyectadas en el espacio h
f	Función del criterio de separabilidad de clases
s	Parámetro de optimización del método paramétrico

Discriminante DTC

$\mathbf{t}$	Punto de tangencia entre las curvas de nivel y el discriminante lineal DTC
α	Relación entre los gradientes de las superficies de nivel en el punto de tangencia que determina un discriminante

Problema MPDH

MPDH	Hiperplano de Decisión Probabilística Minimax (*Minimax Probabilistic Decision Hyperplane*)
ε	Cota inferior de la probabilidad de clasificación correcta

Discriminantes no lineales

m	Número de nodos ocultos
$\tilde{n}$	Número de nodos de salida, $\tilde{n} = 1$ en clasificación binaria
$\mathcal{H}_{\mathrm{NL}}$	Frontera de decisión no lineal
ϕ	Función de transformación no lineal
κ	Función de kernell
L	Matriz de Gram
SLFN	Red Neuronal Propagada hacia adelante de una Sola Capa oculta (*Single-hidden Layer Feedforward Neural Network*)
RBFN	Redes de Funciones de Base Radial (*Radial Basis Function Networks*)

Funciones y operadores

E	Esperanza matemática
∂	Derivada parcial
exp	Función exponencial
P	Probabilidad
pdf	Función de densidad de probabilidad (*probability density function*)
$D_M(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$	Distancia de Mahalanobis entre los puntos $\mathbf{x}^{(1)}$ y $\mathbf{x}^{(2)}$
$\mathcal{O}$	Orden del coste computacional

Introducción

Las máquinas, como los humanos, tienen la capacidad de observar una parte de la realidad (muestras) y, como los humanos nuevamente, usando su memoria (pesos del aprendizaje), hacer una suposición sobre la realidad (inferencia).

La toma de decisiones, de las que nuestras vidas están llenas, tiene su equivalencia en los problemas de clasificación. En el caso más simple, la decisión es elegir a qué situación (clase) de entre dos de ellas pertenece un suceso (clasificación binaria). Analíticamente, el objetivo es determinar qué probabilidad posterior es mayor —$P(C_j|\mathbf{x})$, j=1, 2, dado un suceso $\mathbf{x}$, es la probabilidad de que pertenezca a la clase C_j—. Por lo tanto, la regla de decisión es la razón de las probabilidades posteriores en relación a una política de riesgo o coste que permite establecer cuantitativamente la importancia de haber acertado o errado en la decisión

$$\frac{P(C_2|\mathbf{x})}{P(C_1|\mathbf{x})} \underset{C_1}{\overset{C_2}{\gtrless}} \frac{q_{21}-q_{11}}{q_{12}-q_{22}}\,, \qquad \text{Regla de decisión de mínimo riesgo}$$

siendo q_{dj} el coste de tomar la decisión de que la muestra $\mathbf{x}$ pertenece a la clase C_d cuando la situación real es que pertenece a la clase C_j.

Problema teórico

En un problema teórico en el que se conocen las distribuciones conjuntas *i.e.*, se conocen las densidades de probabilidad de las clases o verosimilitudes $P(\mathbf{x}|C_j)$ y las probabilidades a priori de las clases $P(C_j)$, la solución óptima es la Bayesiana, que minimiza el error de decisión y permite calcular la razón de las probabilidades a posteriori a partir de las verosimilitudes y las probabilidades a priori. La regla de decisión resultante es

$$\frac{P(C_2|\mathbf{x})}{P(C_1|\mathbf{x})} = \frac{P(\mathbf{x}|C_2)P(C_2)}{P(\mathbf{x}|C_1)P(C_1)} \underset{C_1}{\overset{C_2}{\gtrless}} \frac{q_{21}-q_{11}}{q_{12}-q_{22}}\,. \qquad \text{Regla de decisión óptima de Bayes}$$

Problemas reales

En problemas reales, los modelos de las clases $P(\mathbf{x}|C_j)$ son desconocidos y no hay infinitas muestras de datos independientes para estimarlos. Por lo tanto, sin una información a priori completa sobre las clases, no es posible lograr el coste mínimo de decisión de la solución óptima de Bayes. En cambio, si hay muestras etiquetadas disponibles, las distribuciones conjuntas $P(\mathbf{x}, C_j)$ pueden estimarse a partir de las muestras en dos enfoques de aprendizaje supervisado diferentes: los modelos generativos y los modelos discriminativos. Dado que a veces las muestras etiquetadas son costosas de obtener o el número de clases puede ser desconocido, el aprendizaje no supervisado se puede usar de manera independiente o como una primera etapa de clasificación. La agrupación (*clustering*) es el uso principal de los métodos no supervisados.

Modelos generativos

El enfoque generativo estima la distribución conjunta real de las clases –las densidades $P(\mathbf{x}|C_j)$ y las probabilidades a priori C_j– asumiendo la forma de las distribuciones de las clases y estimando sus parámetros a partir de las muestras. De esta manera, las probabilidades posteriores se calculan mediante el teorema de Bayes.

Estos métodos se centran en la representación de las clases de los datos y evalúan la similitud de las nuevas muestras respecto al modelo. Los métodos paramétricos, los grafos y los modelos de Markov son ejemplos de este enfoque. Presentan un rendimiento limitado y el riesgo de elegir un modelo diferente del modelo real.

Modelos discriminativos

Por otro lado, el enfoque discriminativo estima directamente las probabilidades a posteriori $P(C_j|\mathbf{x})$ a partir de las muestras y modela la frontera de decisión de las clases por medio de un coste subrogado que se minimiza. La salida del clasificador hace predicciones basándose en las diferencias entre las clases. Este es el caso de la regresión logística, las máquinas de vectores de soporte (*support vector machines*) o las redes neuronales. En estos métodos, los valores de las variables intermedias de la estructura son difíciles de interpretar. Proporcionan un alto rendimiento aunque con riesgo de sobreajuste.

Análisis discriminante

Entre los modelos discriminativos, el análisis discriminante es un esfuerzo por encontrar una alternativa menos costosa computacionalmente al discriminante de Bayes. En lugar de la estimación de los parámetros de las funciones de densidad, propia de los modelos generativos, se establece el modelo matemático de la frontera de decisión (discriminante) y sus parámetros se estiman a partir de las muestras, como los discriminantes paramétricos del Capítulo 1.

Redes neuronales

Las redes neuronales, dentro de los modelos discriminativos, son adecuadas para resolver problemas más complejos que necesitan fronteras de decisión no lineales. Presentan además capacidades de aproximación universal. Se pueden entrenar de forma supervisada o en una combinación de dos etapas separadas, la primera sin supervisión y la última supervisada, siendo la complejidad del entrenamiento de cada etapa menor que la del entrenamiento conjunto, proporcionando una mejor velocidad de entrenamiento y un menor coste computacional. En esta forma de entrenamiento por etapas, la etapa supervisada se puede entrenar usando el análisis discriminante, como se muestra en el Capítulo 3.

DTC (análisis discriminante y redes neuronales)

El análisis de discriminantes lineales mediante situaciones de tangencia disjunta (DTC), Capítulo 2, se basa en la interpretación geométrica que se deriva de la caracterización de las clases de los datos por sus dos primeros momentos, lo que origina curvas de nivel de probabilidad que son elipsoides, donde los discriminantes corresponden a los hiperplanos tangentes a los distintos elipsoides. DTC pertenece al análisis discriminante y se usa como marco de diseño e interpretación tanto de discriminantes conocidos –tales como los del método paramétrico y los del método de optimización convexa minimax, con los que es posible establecer una correspondencia analítica– como de discriminantes nuevos –como el que se obtiene de la aproximación al óptimo de Bayes, con precisión similar y un coste computacional menor al ser DTC un método no iterativo–. Mediante el uso de redes neuronales de una sola capa oculta con núcleos Gaussianos, Capítulo 3, es posible construir discriminantes DTC no lineales a partir de los DTCs lineales, con un coste computacional reducido debido a la posibilidad de pre-entrenar de manera no supervisada los núcleos Gaussianos. De la interpretación geométrica del DTC se deriva de forma natural la solución minimax, tanto en los discriminantes lineales como en los no lineales, adecuada para problemas en los que se desea minimizar el riesgo asociado al desconocimiento de las distribuciones de las clases de los datos o sus probabilidades *a priori*.

Parte I

Discriminantes DTC lineales

Dado un conjunto de muestras o patrones que pertenecen a dos clases (problema binario), existe la necesidad de decidir (**clasificación**) a qué clase pertenecen nuevas muestras cometiendo el menor error posible. Un posible clasificador es el discriminante lineal, *i.e.*, un hiperplano en el caso genérico de múltiples dimensiones que separa el espacio en dos regiones, una por clase, permitiendo decidir a cuál de ellas pertenece una nueva muestra. Aunque los discriminantes lineales sólo son óptimos para clases distribuidas normalmente con matrices de covarianza iguales, en muchos casos se prefieren por su simplicidad, robustez y fácil interpretación.

El clasificador de **Bayes** es óptimo porque minimiza la probabilidad de error (Fukunaga, 1990), pero requiere el conocimiento de las funciones de densidad de probabilidad de las clases a partir de técnicas de estimación que son complejas computacionalmente y necesitan grandes cantidades de datos para proporcionar resultados precisos.

En consecuencia, se han desarrollado procedimientos más simples como las **técnicas paramétricas**, que especifican la forma matemática del clasificador seguido de la estimación de sus parámetros. Tal es el caso del procedimiento de Fukunaga (1990) de diseño de discriminantes lineales para problemas binarios. Este método es óptimo respecto a un **criterio de separabilidad** definido en un espacio proyectado unidimensional, *i.e.*, una dirección a lo largo de la cual los datos proyectados de una clase son máximamente separados de los datos proyectados de la otra. Los distintos discriminantes lineales se obtienen a partir de diferentes criterios de separabilidad. Los criterios más importantes son el error de Bayes para distribuciones normales, el criterio de Fisher (1923) y otros criterios basados en matrices de dispersión (*scatter*) (Duda et al., 2000).

Por último, se presenta una nueva interpretación de los discriminantes lineales en la que son descritos en términos de Situaciones de Tangencia Disjunta (*Disjoint Tangent Configurations*, **DTC**) por Sancho-Gómez, Martínez-García, Ahalt y Figueiras-Vidal (2018), cuyas fronteras de decisión son los hiperplanos tangentes a las diferentes superficies de nivel de probabilidad (elipsoides) definidas por los dos primeros momentos de las distribuciones de las clases. Es posible establecer una **correspondencia analítica** entre la formulación del método paramétrico y la interpretación de sus discriminantes como situaciones de tangencia disjunta (DTC). También es posible establecer una correspondencia con los discriminantes obtenidos por el método de optimización convexa (Bertsimas y Popescu, 2005) basado en minimizar la máxima probabilidad de error, tales como la Máquina de Probabilidad Minimax (*Minimax Probability Machine*, MPM) de Lanckriet et al. (2002), que es la solución del problema del Hiperplano de Decisión Probabilística Minimax (*Minimax Probabilistic Decision Hyperplane*, MPDH), y la Máquina de Probabilidad Minimax de Mínimo Error (*Minimum Error Minimax Probability Machine*, MEMPM) de Huang et al. (2004), que proporciona una solución Bayesiana. DTC proporciona una **solución directa** con un coste computacional competitivo, en términos de de velocidad y memoria, en contraste con el método de optimización convexa, que es iterativo. Este nuevo marco de diseño también permite obtener **nuevos discriminantes** con propiedades interesantes, en particular, el discriminante Quasi-Bayes, que proporciona una precisión cercana a la del discriminante lineal de Bayes con la ventaja de necesitar un coste computacional menor.

1

Diseño de discriminantes lineales

1.1 *Diseño paramétrico*

El método paramétrico de Fukunaga (1990) es un procedimiento simple de diseño de clasificadores lineales binarios en el que se especifica la forma matemática del clasificador, que incluye un conjunto de parámetros libres que necesitan ser ajustados en un proceso de optimización, *i.e.*, los vectores de pesos y sesgos de la función del discriminante lineal $h : \mathbb{R}^n \to \mathbb{R}$, que se expresa de la siguiente forma

$$h(\mathbf{x}) = \mathbf{w}^T(\mathbf{x} - \mathbf{x}_0) \,, \tag{1.1}$$

donde $\mathbf{x} \in \mathbb{R}^n$ son las muestras de entrada, $\mathbf{w} \in \mathbb{R}^n$ el vector de pesos y $\mathbf{x}_0 \in \mathbb{R}^n$ el vector de sesgos y un punto cualquiera del discriminante.

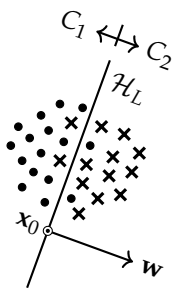


Figura 1.1: Ejemplo de discriminante lineal con muestras de la clase C_1 (puntos) y de la clase C_2 (cruces).

REGLA DE DECISIÓN

En consecuencia, la regla de decisión para un problema de clasificación de dos clases es

$$h(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + \omega_0 \underset{C_2}{\overset{C_1}{\gtrless}} 0 \,, \tag{1.2}$$

donde

$$\omega_0 = -\mathbf{w}^T\mathbf{x}_0 \,, \tag{1.3}$$

y se interpreta como la proyección de las muestras en la dirección del vector $\mathbf{w}$, que son clasificadas como pertenecientes a la clase C_1 o la clase C_2 dependiendo de si la variable $z = \mathbf{w}^T\mathbf{x}$ es mayor o menor que $-\omega_0$, llamado valor umbral. Por lo tanto, $h(\mathbf{x})$ es conocido también como el espacio de proyección unidimensional o la función de proyección de las muestras en $\mathbf{w}$.

FRONTERA DE DECISIÓN

La ecuación $h(\mathbf{x}) = 0$ describe la frontera de decisión, que corresponde en el caso general n-dimensional al hiperplano $\mathcal{H}_L(\mathbf{w}, \omega_0) = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{w}^T\mathbf{x} + \omega_0 = 0\}$ en el que el vector de pesos $\mathbf{w}$ determina su orientación y el vector de sesgos $\mathbf{x}_0$ fija su posición relativa en el espacio de los datos, ver Figura 1.1. De esta forma, un discriminante lineal divide por medio de un hiperplano el espacio en dos espacios-mitad correspondientes a las regiones de decisión.

DISEÑO. Encontrar $\mathbf{w}$ y ω_0 que presenten menor error de clasificación según el criterio de separabilidad f.

El diseño de un clasificador lineal consiste en encontrar el vector

de pesos $\mathbf{w}$ y el valor umbral ω_0 (o el vector de sesgos $\mathbf{x}_0$) óptimos que proporcionen el menor error de clasificación en el espacio proyectado unidimensional h como resultado de la optimización de un determinado criterio de separabilidad de las clases. Es decir, primero se escoge un criterio de separabilidad a través de la función f que mida el grado de separación de las clases, y después se busca la dirección de proyección $\mathbf{w}$ que, según el criterio de separabilidad escogido, presente menor error de clasificación. La Figura 1.2 muestra cómo la dirección de proyección $\mathbf{w}$ presenta menor error de clasificación (zona sombreada) que la dirección $\mathbf{w}'$ en dos clases representadas por sus medias y matrices de covarianza.

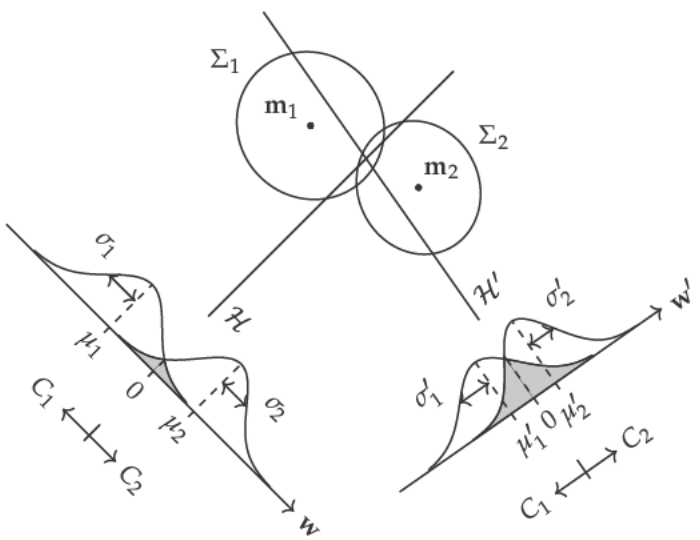


Figura 1.2: Espacio proyectado h en dos direcciones distintas, $\mathbf{w}$ y $\mathbf{w}'$, de los discriminantes lineales.

Cuando $\mathbf{x}$ está distribuido normalmente o n es elevado, $h(\mathbf{x})$ es también normal o casi normal, debido al teorema del límite central, respectivamente. En este caso, el criterio apropiado f para medir la separabilidad de las clases depende de las medias y varianzas de $h(\mathbf{x})$, *i.e.*, $f(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2)$, que son

$$\begin{aligned}\mu_j &= E\{h(\mathbf{x})|C_j\} = \mathbf{w}^T E\{\mathbf{x} \mid C_j\} + \omega_0 \\ &= \mathbf{w}^T \mathbf{m}_j + \omega_0 \,, \qquad (1.4a)\\ \sigma_j^2 &= Var\{h(\mathbf{x})|C_j\} = \mathbf{w}^T E\{(\mathbf{x}-\mathbf{m}_j)(\mathbf{x}-\mathbf{m}_j)^T \mid C_j\}\,\mathbf{w} \\ &= \mathbf{w}^T \Sigma_j \mathbf{w} \,, \qquad (1.4b)\end{aligned}$$

donde $\mathbf{m}_j \in \mathbb{R}^n$ y $\Sigma_j \in \mathbb{R}^{n\times n}$, j=1, 2, son respectivamente el vector media y la matriz de covarianza de las muestras de cada clase antes de proyectar. Como se mencionó antes, los valores óptimos de los parámetros $\mathbf{w}$ y ω_0 se obtienen a partir de la optimización de la función del criterio de separabilidad f (Fukunaga, 1990), resultando

$$\mathbf{w}^{\mathrm{P}} = \left[s\Sigma_1 + (1-s)\Sigma_2\right]^{-1}(\mathbf{m}_2 - \mathbf{m}_1) \,, \qquad (1.5)$$

CRITERIO DE SEPARABILIDAD. Es función las medias y varianzas en el espacio proyectado.

TEOREMA DEL LÍMITE CENTRAL: las medias de diferentes muestras de una distribución se aproximan a una distribución normal conforme el tamaño de la muestra crece.

De esta forma, el producto escalar de un patrón por el vector de pesos, $z_i = \mathbf{w}^T\mathbf{x}^{(i)} = w_1 x_1^{(i)} + w_2 x_2^{(i)} + \ldots + w_n x_n^{(i)}$, i=1, 2, ..., N, se puede considerar una "media ponderada" de las componentes de $\mathbf{x}^{(i)}$, siendo los patrones proyectados $\{z_1, z_2, \ldots, z_N\}$, aproximadamente una distribución normal si n es suficientemente grande.

donde

$$s = \frac{\partial f/\partial \sigma_1^2}{\partial f/\partial \sigma_1^2 + \partial f/\partial \sigma_2^2} \,, \tag{1.6}$$

y ω_0^{P}, que se obtiene como la solución de

$$\frac{\partial f}{\partial \mu_1} + \frac{\partial f}{\partial \mu_2} = 0 \,. \tag{1.7}$$

MÉTODO PARAMÉTRICO $(.)^{\mathrm{P}}$
Crit. separab. $f(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2)$
SOLUCIÓN GENERAL:
$\mathbf{w}^{\mathrm{P}}(s)$ (1.5), $\omega_0^{\mathrm{P}}(s)$ (1.7)

Obsérvese que $\mathbf{w}^{\mathrm{P}}$ (1.5) no depende de la selección de f, que sólo afecta a s (1.6). Es más, esta solución no sólo es válida para distribuciones condicionales de clase normales, sino también para cualquier problema de clasificación binaria en el que las medias y las matrices de covarianza sean conocidas.

Los diferentes discriminantes lineales corresponden a diferentes selecciones de f, algunos de ellos son presentados a continuación.

1.1.1 *Discriminante de Bayes para distribuciones Gaussianas*

Como se mencionó anteriormente, en el caso de distribuciones normales ($\mathbf{x} \sim N(\mathbf{m}_j, \Sigma_j)$), $h(\mathbf{x})$ es también normal y, por lo tanto, el error de clasificación en el espacio h es el error de Bayes. El discriminante lineal de Bayes tiene por objetivo minimizar el error de clasificación, *i.e.*, el criterio de separabilidad es el error de Bayes. La búsqueda del mínimo error de Bayes se puede realizar a través de un procedimiento iterativo descrito más adelante.

Para determinar los parámetros del discriminante, el error de Bayes se puede expresar en función de μ_j y σ_j^2 como

$$f = \frac{P_1}{\sqrt{2\pi}} \int_{-\frac{\mu_1}{\sigma_1}}^{+\infty} \exp\left(\frac{-\xi^2}{2}\right) d\xi + \frac{P_2}{\sqrt{2\pi}} \int_{-\infty}^{-\frac{\mu_2}{\sigma_2}} \exp\left(\frac{-\xi^2}{2}\right) d\xi \,, \tag{1.8}$$

donde P_j, j=1, 2, es la probabilidad *a priori* de la clase j-ésima.

Se puede probar (Fukunaga, 1990) que, en este caso, el discriminante lineal óptimo es dado por $\mathbf{w}^{\mathrm{P}}$ (1.5) con

$$s = \frac{-\mu_1/\sigma_1^2}{-\mu_1/\sigma_1^2 + \mu_2/\sigma_2^2} \,, \tag{1.9}$$

siendo $0 < s < 1$ ya que $\mu_1 < 0$ y $\mu_2 > 0$. Por otra parte, ω_0^{PB} se obtiene como la solución de (1.7), que resulta la siguiente igualdad

$$\frac{P_1}{\sigma_1\sqrt{2\pi}} \exp\left(\frac{-\mu_1^2}{2\sigma_1^2}\right) = \frac{P_2}{\sigma_2\sqrt{2\pi}} \exp\left(\frac{-\mu_2^2}{2\sigma_2^2}\right) . \tag{1.10}$$

MÉTODO PARAMÉTRICO
DISC. BAYES LINEAL $(.)^{\mathrm{PB}}$
f: error de Bayes (1.8)
SOLUCIÓN (no explícita):
$\mathbf{w}^{\mathrm{P}}(s^*)$ (1.5)
$\omega_0^{\mathrm{P}}(s^*)$ (1.7) $\rightarrow$ $\omega_0^{\mathrm{PB}}(s^*)$ (1.11)
s^*: óptimo (error de Bayes mínimo)

La expresión de ω_0^{PB} en función de s y $\mathbf{w}^{\mathrm{PB}}$ se obtiene fácilmente sustituyendo μ_1 y μ_2 de (1.4a) en (1.9) y despejando ω_0^{PB}

$$\omega_0^{\mathrm{PB}} = -\frac{s\sigma_1^2(\mathbf{w}^{\mathrm{PB}})^T\mathbf{m}_2 + (1-s)\sigma_2^2(\mathbf{w}^{\mathrm{PB}})^T\mathbf{m}_1}{s\sigma_1^2 + (1-s)\sigma_2^2} \,. \tag{1.11}$$

Adviértase que s en (1.9) es función de $\mathbf{w}^{\mathrm{P}}$ y ω_0^{P} debido a (1.4), y $\mathbf{w}^{\mathrm{P}}$ en (1.5) es función de s; por lo tanto, no se puede obtener de esta forma una solución óptima explícita para $\mathbf{w}^{\mathrm{PB}}$ y ω_0^{PB} debido a su interdependencia. Sin embargo, existe un procedimiento iterativo para hallar el discriminante lineal de Bayes llamado el Método Teórico (TM) (Fukunaga, 1990; Peterson y Mattson, 1966), que se muestra a continuación

Algoritmo 1: Método Teórico (TM)

Datos: P_j, $\mathbf{m}_j$, Σ_j, j=1,2
Resultado: $\mathbf{w}^*$, ω_0^*
for *(barrido de* $s \in [0,1]$ *con pasos* Δs *)* **do**
  $\mathbf{w} \leftarrow s$ en (1.5)
  $\sigma_j^2 \leftarrow \mathbf{w}$ en (1.4b)
  $\omega_0 \leftarrow \mathbf{w}$ y σ_j^2 en (1.11)
  $\mu_j \leftarrow \mathbf{w}$ y ω_0 en (1.4a)
  $f \leftarrow \sigma_j^2$ y μ_j en (1.8)
  if *f es mínimo* **then**
    $\mathbf{w}^* \leftarrow \mathbf{w}$
    $\omega_0^* \leftarrow \omega_0$

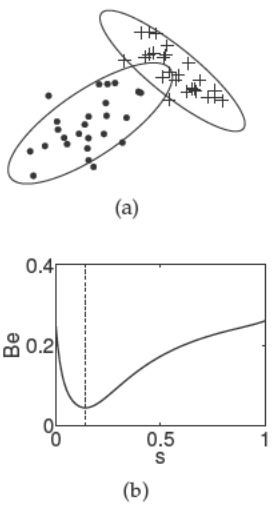


Figura 1.3: (a) Curvas de nivel correspondientes a dos poblaciones normales bidimensionales. (b) Los resultados de la simulación del Método Teórico (TM) (error de Bayes (Be) vs. s y el valor mínimo en s^*).

A pesar de que es un proceso fácil y eficiente, es un procedimiento cuyo resultado depende del paso Δs. La Figura 1.3 lo ilustra, se observa cómo Δs tiene que ser suficientemente pequeño para asegurar un discriminante satisfactorio.

1.1.2 *Discriminante de Fisher*

El criterio de separabilidad de Fisher (1923) viene dado por

$$f = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}\,, \tag{1.12}$$

que mide la diferencia entre las dos medias normalizada por la varianza promedio. Tomando derivadas parciales de f respecto a σ^2

$$\frac{\partial f}{\partial \sigma_1^2} = \frac{\partial f}{\partial \sigma_2^2} = \frac{-(\mu_1 - \mu_2)^2}{(\sigma_1^2 + \sigma_2^2)^2}\,, \tag{1.13}$$

y sustituyendo en (1.6) se obtiene como resultado s=0.5. Así, el $\mathbf{w}$ óptimo de (1.5) queda de la siguiente forma

$$\mathbf{w}^{\mathrm{PF}} = \left[\frac{1}{2}\Sigma_1 + \frac{1}{2}\Sigma_2\right]^{-1}(\mathbf{m}_2 - \mathbf{m}_1)\,. \tag{1.14}$$

MÉTODO PARAMÉTRICO
DISC. FISHER $(.)^{\mathrm{PF}}$
f (1.12)
SOLUCIÓN:
$\mathbf{w}^{\mathrm{P}}(s{=}0.5)$ (1.5) $\rightarrow \mathbf{w}^{\mathrm{PF}}$ (1.14)
sin ω_0

Este criterio de separabilidad no depende de ω_0 porque la resta de μ_2 a μ_1 en (1.12) a partir de (1.4a) elimina ω_0, de modo que no puede ser calculado maximizando f. Por consiguiente, la solución de Fisher no es un discriminante completo sino solamente una dirección de

proyección óptima. Por supuesto, hay procedimientos complementarios para determinar ω_0^{PF}, tales como usar la equivalencia con una distribución normal o maximizar la separación de los valores de las proyecciones de las muestras.

1.1.3 *Discriminante basado en Scatter*

Otro criterio importante de separabilidad de clases es

$$f = \frac{P_1\mu_1^2 + P_2\mu_2^2}{P_1\sigma_1^2 + P_2\sigma_2^2} \,, \tag{1.15}$$

que mide la dispersión (*scatter*, alrededor de cero) de interclase normalizada por la dispersión intraclase (Fukunaga, 1990). Tomando derivadas parciales de f respecto a σ_1^2 y σ_2^2

$$\frac{\partial f}{\partial \sigma_i^2} = \frac{-P_i(P_1\mu_1^2 + P_2\mu_2^2)}{(P_1\sigma_1^2 + P_2\sigma_2^2)^2} \,, \tag{1.16}$$

y sustituyendo en s (1.6) se obtiene como resultado $s = P_1$. Así, el $\mathbf{w}^{\text{P}}$ óptimo de (1.5) queda de la siguiente forma

$$\mathbf{w}^{\text{PS}} = \left[P_1\Sigma_1 + P_2\Sigma_2\right]^{-1}(\mathbf{m}_2 - \mathbf{m}_1) \,. \tag{1.17}$$

MÉTODO PARAMÉTRICO
DISC. SCATTER $(.)^{\text{PS}}$
f (1.15)
SOLUCIÓN:
$\mathbf{w}^{\text{P}}(s = P_1)$ (1.5) $\rightarrow \mathbf{w}^{\text{PS}}$ (1.17)
$\omega_0^{\text{P}}(s = P_1)$ (1.7) $\rightarrow \omega_0^{\text{PS}}$ (1.19)

ω_0^{P} se obtiene tomando derivadas parciales de f respecto a μ_1^2 y μ_2^2

$$\frac{\partial f}{\partial \mu_i} = \frac{2P_i\mu_i}{P_1\sigma_1^2 + P_2\sigma_2^2} \,, \tag{1.18}$$

y sustituyendo en (1.7) con (1.4a) resulta

$$\omega_0^{\text{PS}} = -(\mathbf{w}^{\text{PS}})^T\left[P_1\mathbf{m}_1 + P_2\mathbf{m}_2\right] \,, \tag{1.19}$$

que muestra cómo si $\mathbf{w}^{\text{PS}}$ se multiplica por una constante, ω_0^{PS} cambia en el mismo factor y, por tanto, la frontera de decisión del discriminante es la misma.

1.2 *Diseño mediante optimización convexa minimax*

Acotar y minimizar el error

El método de optimización convexa minimax hace uso de las desigualdades de Marshall y Olkin (1960) basadas en los momentos de las distribuciones de las clases de los datos y consiste en minimizar la probabilidad de clasificación errónea sin hacer suposiciones sobre las distribuciones de las densidades condicionales de las clases, ya que no ofrece suficiente generalidad ni validez. Por lo tanto, primero, la probabilidad de clasificación errónea se acota a partir de los dos primeros momentos de las distribuciones de las clases de los datos, como se muestra en el Apéndice A, y después se minimiza. De esta manera, la solución es válida para todas las posibles elecciones de las densidades condicionales de las clases con una determinada media y matriz de covarianza.

Así, el problema general de optimización se puede expresar de la siguiente forma

PROBLEMA GENERAL DE OPTIMIZACIÓN MINIMAX

$$\begin{aligned} \max_{\varepsilon_1,\varepsilon_2,\mathbf{w}\neq\mathbf{0},\omega_0} \quad & \lambda_1\varepsilon_1 + \lambda_2\varepsilon_2 \quad s.t. \\ & \inf_{\mathbf{x}\sim(\mathbf{m}_1,\Sigma_1)} P\{\mathbf{w}^T\mathbf{x}+\omega_0 \geq 0\} \geq \varepsilon_1 \\ & \inf_{\mathbf{x}\sim(\mathbf{m}_2,\Sigma_2)} P\{\mathbf{w}^T\mathbf{x}+\omega_0 \leq 0\} \geq \varepsilon_2 \ , \end{aligned} \tag{1.20}$$

donde $\mathbf{x} \sim (\mathbf{m}_j, \Sigma_j)$ representa todas las distribuciones arbitrarias con la misma media $\mathbf{m}_j$ y matriz de covarianza Σ_j; $\lambda_j \in \mathbb{R}$ y $\varepsilon_j \in \mathbb{R}$ son constantes, ε_j acota la probabilidad clasificación correcta sobre todas las distribuciones descritas. Para simplificar, considérese el caso de igual ε para ambas clases y, por lo tanto, sin necesidad de las constantes λ_j. Para minimizar el máximo error de clasificación, se maximiza la cota inferior ε de la probabilidad de acierto del discriminante, variando el propio ε y los parámetros del discriminante ($\mathbf{w}$, ω_0), sujeta a la condición de que el ínfimo de las probabilidades de clasificación correcta de las distribuciones que comparten la misma media $\mathbf{m}_j$ y matriz de covarianza Σ_j sea mayor o igual que ε.

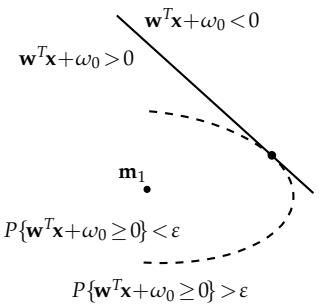


Figura 1.4: Acotación de la probabilidad de clasificación correcta en el método de optimización.

Por tanto, al maximizar la mínima probabilidad de acierto ε, debido al principio de dualidad, se minimiza la máxima probabilidad de error $(1-\varepsilon)$ –minimización del peor caso–, quedando acotada inferiormente la probabilidad de acierto y superiormente la probabilidad de error. Puesto que el error de clasificación depende de la distribución de los datos y esta solución es válida para todas las distribuciones arbitrarias con los mismos dos primeros momentos, minimizar el máximo riesgo significa encontrar el discriminante que minimice el error producido por la distribución que mayor error presentaría, lo que define el criterio minimax.

Solución minimax

La solución minimax, Apéndice B, es una cuestión importante en casos tales como cuando el número de datos de entrenamiento de cada clase no refleja las verdaderas probabilidades *a priori*. Por lo tanto, minimax es un criterio de clasificación natural en ausencia de información *a priori* sobre la frecuencia real de las dos clases. Por esta razón, bastantes investigadores prefieren usar clasificadores que operen a igual tasa de error (*Equal Error Rate*, EER), esto es, clasificadores que minimicen el máximo de las falsas alarmas y las tasas de fallos (Sebastiani, 2002; Bengio et al., 2005). Es más, el problema minimax puede ser también abordado cuando la información de las clases es desconocida o no es exacta. Adviértase que (1.20) usa cotas, y consecuentemente su solución no tiene que satisfacer la condición EER, *i.e.*, es una solución aproximada.

Adviértase que ε es también un criterio de separabilidad f, incluso aunque no tenga la forma $f(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2)$ y, por lo tanto, la solución se puede expresar en términos de (1.5)-(1.6). En la Figura 1.4 se muestra ε, la cota inferior de la probabilidad de acierto (región izquierda del discriminante), con línea discontinua. Conforme crece

el valor de ε, la distancia de Mahalanobis d respecto a la media $\mathbf{m}_1$ aumenta, lo que supone agrandar la línea discontinua hacia la derecha. El valor máximo que puede alcanzar ε, o la superficie de nivel de probabilidad asociada, que es un elipsoide en el caso general, es tangente a la frontera del discriminante. El discriminante es común para ambas clases y también se va modificando en el proceso de maximización de ε, en caso contrario, el discriminante quedaría más cerca de alguna de las dos clases y para la otra clase el valor de ε no sería óptimo. En la solución, el discriminante es tangente a las dos elipses de probabilidad. Las elipses crecen a velocidad $\kappa(\varepsilon) = \sqrt{\frac{\varepsilon}{1-\varepsilon}}$.

Optimización convexa. Solución iterativa

En Lanckriet et al. (2002) y Huang et al. (2004), el problema de optimización (1.20) se aborda con el Programa de Cono de Segundo Orden (*Second-Order Cone Program*, SOCP) de Boyd y Vandenberghe (2004), resolviendo por medio de una aproximación iterativa de mínimos cuadrados. También incluye un proceso regulador de la matriz Hessiana para incrementar la estabilidad computacional y resuelve el problema de la robustez respecto a los errores de estimación de las medias y matrices de covarianza mediante la regularización de los datos de entrada.

1.2.1 *Discriminante de Bayes para distribuciones Gaussianas*

La solución de la Máquina de Probabilidad Minimax de Mínimo Error (*Minimum Error Minimax Probability Machine*, MEMPM) es proporcionada por Huang et al. (2004) como la solución de la optimización (1.20) maximizando la probabilidad de acierto de ambas clases ε_j, con $\lambda_j = P_j$, j=1, 2, como sigue

MEMPM (método de optimización minimax)

$$\begin{aligned} \max_{\varepsilon_1,\varepsilon_2,\mathbf{w}\neq\mathbf{0},\omega_0} \quad & P_1\varepsilon_1 + (1-P_1)\varepsilon_2 \quad s.t. \\ & \inf_{\mathbf{x}\sim(\mathbf{m}_1,\Sigma_1)} P\{\mathbf{w}^T\mathbf{x} + \omega_0 \geq 0\} \geq \varepsilon_1 \\ & \inf_{\mathbf{x}\sim(\mathbf{m}_2,\Sigma_2)} P\{\mathbf{w}^T\mathbf{x} + \omega_0 \leq 0\} \geq \varepsilon_2 \,. \end{aligned} \tag{1.21}$$

MEMPM minimiza el peor caso (comportamiento minimax) del error de Bayes. De esta manera, MEMPM se convierte, bajo condiciones de Gaussianidad, en el discriminante óptimo de Bayes, mientras que en un caso general es una aproximación.

1.2.2 *Discriminante MPDH*

Lanckriet et al. (2002) proponen la Máquina de Probabilidad Minimax (*Minimax Probability Machine*, MPM) para clasificación binaria como solución del problema del Hiperplano de Decisión Probabilística Minimax (*Minimax Probabilistic Decision Hyperplane*, MPDH), que es la solución del problema de optimización (1.20) cuando las

probabilidades de acierto de cada clase son iguales, $\varepsilon_1 = \varepsilon_2 = \varepsilon$,

$$\begin{aligned} \max_{\varepsilon, \mathbf{w} \neq \mathbf{0}, \omega_0} \ \varepsilon \quad s.t. \quad & \inf_{\mathbf{x} \sim (\mathbf{m}_1, \Sigma_1)} P\{\mathbf{w}^T \mathbf{x} + \omega_0 \geq 0\} \geq \varepsilon \\ & \inf_{\mathbf{x} \sim (\mathbf{m}_2, \Sigma_2)} P\{\mathbf{w}^T \mathbf{x} + \omega_0 \leq 0\} \geq \varepsilon \ . \end{aligned} \tag{1.22}$$

MPDH (método de optimización minimax)

Como se ha mencionado, se maximiza la cota inferior de la probabilidad de clasificación correcta, para cualquier distribución arbitraria con la misma media y matriz de covarianza, para minimizar el peor caso, la máxima probabilidad clasificación errónea.

Lanckriet et al. (2002) prueban que los parámetros ε_* y $\mathbf{w}_*$ de la solución de (1.22) están relacionados por la siguiente ecuación

$$1 - \varepsilon_* = \frac{\left(\sqrt{\mathbf{w}_*^T \Sigma_1 \mathbf{w}_*} + \sqrt{\mathbf{w}_*^T \Sigma_2 \mathbf{w}} \right)^2}{1 + \left(\sqrt{\mathbf{w}_*^T \Sigma_1 \mathbf{w}_*} + \sqrt{\mathbf{w}_*^T \Sigma_2 \mathbf{w}_*} \right)^2} \ , \tag{1.23}$$

ε_* se puede obtener de esta ecuación cuando el MPDH óptimo es encontrado por el algoritmo MPM o cualquier otro procedimiento de optimización.

Adivértase que MPM es un caso particular de MEMPM.

2

Discriminantes de situaciones de tangencia disjunta (DTC)

El método de discriminantes basados en Situaciones de Tangencia Disjunta (*Disjoint Tangent Configurations*, DTC), (Sancho-Gómez et al., 2018), establece una nueva interpretación de los discriminantes lineales para clasificación binaria. Como en el método de optimización convexa (Bertsimas y Popescu, 2005) de MPM (Lanckriet et al., 2002) y MEMPM (Huang et al., 2004), las distribuciones de las clases también están caracterizadas por sus dos primeros momentos, el vector media $\mathbf{m}$ y la matriz de covarianza Σ, lo que produce superficies de nivel de probabilidad que son elipsoides. De esta forma, los discriminantes DTC quedan definidos por los puntos de tangencia entre distintos elipsoides. La Figura 2.1 muestra dos posibles puntos de tangencia, $\mathbf{t}_1$ y $\mathbf{t}_2$, en dos situaciones distintas de tangencia entre elipsoides –elipses en el caso bidimensional– y los discriminantes lineales DTC asociados, cuyas fronteras de decisión son las rectas tangentes a los elipsoides que pasan por los puntos de tangencia ya mencionados. Las distribuciones de las clases de los datos no están limitadas a las Gaussianas y pueden ser desconocidas, sólo es necesario una estimación de las medias y de las matrices de covarianza a partir de las muestras.

MÉTODO DTC: elipsoides tangentes

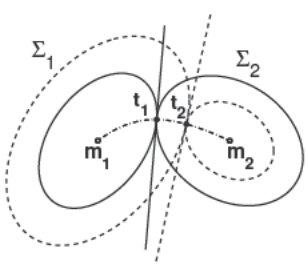


Figura 2.1: Ejemplo bidimensional de dos discriminantes lineales DTC (líneas continua y a trazos), los puntos de tangencia $\mathbf{t}_1$ y $\mathbf{t}_2$ y la curva de todos los puntos de tangencia posibles (curva de trazos y puntos entre las medias).

Este método tiene tres ventajas principales: Primero, es un marco que proporciona una interpretación común de los discriminantes lineales con una correspondencia analítica con el método paramétrico y el método de optimización convexa; segundo, no es un proceso de optimización iterativa, como el método de optimización de MPM y MEMPM, sino un método directo para calcular los elipsoides y sus puntos de tangencia con un coste computacional competitivo en términos de velocidad y uso de memoria; y tercero, ofrece nuevas soluciones como la derivada de la interpretación geométrica del discriminante óptimo cuadrático de Bayes en relación al DTC, llamada Quasi-Bayes ya que obtiene una precisión similar a la Bayesiana con menor coste computacional, o un determinante completo de Fisher, que incluye el término independiente.

Ventajas de DTC

- Marco de interpretación común
- Coste competitivo
- Diseño de nuevos discriminantes

2.1 *Expresión analítica de los discriminantes DTC*

Caracterización de las clases por sus dos primeros momentos. Elipsoides tangentes

Para establecer la forma analítica de los discriminantes DTC considérense las superficies de nivel de probabilidad de dos distribuciones normales $p_j(\mathbf{x})$, j=1, 2, descritas por los vectores de media $\mathbf{m}_j \in \mathbb{R}^n$ y las matrices de covarianza $\Sigma_j \in \mathbb{R}^{n\times n}$ que caracterizan las distribuciones de las clases de los datos. Se pueden obtener diversos puntos tangentes entre las diferentes superficies de nivel de $p(\mathbf{x})$ de cada clase, que son elipsoides debido a que están definidas por los dos primeros momentos. La frontera de decisión de cada discriminante lineal DTC es el hiperplano que pasa a través del punto tangente entre un par de elipsoides y es también tangente a ellos.

Una condición necesaria y suficiente para un DTC es que los vectores gradiente de las densidades condicionales de clase en el punto tangente $\mathbf{t}$ sean paralelos y tengan direcciones opuestas,

$$\nabla p_1(\mathbf{t}) = \beta \cdot \nabla p_2(\mathbf{t}) \,, \tag{2.1}$$

donde β es un número real negativo. Por tanto, consistentemente con la expresión de un discriminante lineal genérico (1.1), la función del discriminante asociada con un DTC se puede expresar como

$$h^{\text{DTC}}(\mathbf{x}) = (\mathbf{w}^{\text{DTC}})^T\mathbf{x} + \omega_0^{\text{DTC}} \,, \tag{2.2}$$

donde

$$\mathbf{w}^{\text{DTC}} = \nabla p_j(\mathbf{t}) \,, \tag{2.3a}$$
$$\omega_0^{\text{DTC}} = -(\mathbf{w}^{\text{DTC}})^T\mathbf{t} \,, \tag{2.3b}$$

escogiendo el punto de tangencia como vector de sesgos $\mathbf{x}_0^{\text{DTC}} = \mathbf{t}$ (recuérdese que cualquier vector del discriminante lineal es válido como vector de sesgo) y el gradiente de las densidades condicionales en $\mathbf{t}$ como el vector de pesos $\mathbf{w}$. La Figura 2.2 muestra el punto de tangencia, los vectores gradiente y la frontera de decisión lineal asociada con el DTC.

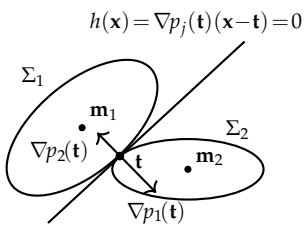
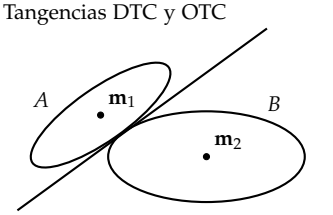


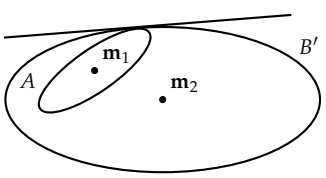
Figura 2.2: Discriminante lineal basado en DTC. La frontera de decisión $h(\mathbf{x}) = 0$, el punto de tangencia $\mathbf{t}$ y el vector gradiente de las superficies de nivel en $\mathbf{t}$, $\nabla p_j(\mathbf{t})$, j=1, 2.

Tangencias DTC y OTC

En la Figura 2.3 se observan los dos tipos de situaciones de tangencia entre elipses bidimensionales. Para una curva de nivel fija de la clase C_1 (elipse etiquetada con A) pueden ocurrir tanto una situación de tangencia disjunta (*Disjoint Tangent Configuration*, DTC) como una situación de tangencia solapada (*Overlapping Tangent Configuration*, OTC) con la clase C_2. En el primer caso, la curva de nivel de probabilidad de la clase C_2 (etiquetada con B) es tangente y disjunta a la curva A, *i.e.*, la intersección de las áreas que encierran cada una es nula. En el segundo caso, la curva de nivel A es encerrada por la curva de nivel correspondiente a C_2 (etiquetada con B'). Puesto que el objetivo es discriminar patrones, sólo se considera la situación DTC ya que OTC no es de utilidad.



(a) DTC



(b) OTC

Figura 2.3: Tipos de tangencia.

Adviértanse dos hechos. Primero, dada una superficie de nivel de probabilidad de la clase C_1, existe una única superficie de nivel

de C_2 que es tangente y disjunta a la anterior. Segundo, cada DTC determina un único hiperplano que es tangente a ambos elipsoides y pasa a través de su punto de tangencia.

Cálculo del punto de tangencia

Puesto que para caracterizar las clases de los datos C_j, $j=1,2$, DTC sólo considera sus dos primeros momentos, se usa la siguiente densidad de probabilidad normal para calcular el punto de tangencia DTC,

$$p_j(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_j|}} \exp\left(-\frac{1}{2} D_M^2(\mathbf{x}, \mathbf{m}_j)\right) , \tag{2.4}$$

donde

$$D_M(\mathbf{x}, \mathbf{m}_j) = \sqrt{(\mathbf{x}-\mathbf{m}_j)^T \Sigma_j^{-1} (\mathbf{x}-\mathbf{m}_j)} \tag{2.5}$$

es la distancia de Mahalanobis de un punto cualquiera $\mathbf{x}$ a la media $\mathbf{m}_j$ de cada clase. Las curvas de nivel de $p_j(\mathbf{x})$ son elipsoides y están definidas por

$$p_j(\mathbf{x}) = c_j , \tag{2.6}$$

siendo c_j real positivo, o equivalentemente

$$D_M(\mathbf{x}, \mathbf{m}_j) = r_j , \tag{2.7}$$

Para cada valor r_j, (2.7) representa un elipsoide E_j de dimensión n que encierra una masa de probabilidad. La masa de probabilidad de una región S se define como la probabilidad de que un patrón $\mathbf{x}$ perteneciente a una distribución $p_j(\mathbf{x})$ caiga dentro de S. Por tanto,

$$E_j \triangleq \{\mathbf{x} : D_M(\mathbf{x}, \mathbf{m}_j) = r_j\} \tag{2.8}$$

con $r_j = \sqrt{-\ln\left(c_j^2 (2\pi)^n |\Sigma_j|\right)}$, siendo $r_j > 0$.

Aplicando el gradiente a $p_j(\mathbf{x})$ en (2.4), se obtiene

$$\nabla p_j(\mathbf{x}) = -p_j(\mathbf{x}) \cdot D_M(\mathbf{x}, \mathbf{m}_j) \cdot \nabla D_M(\mathbf{x}, \mathbf{m}_j) . \tag{2.9}$$

Sustituyendo (2.9) en (2.1) en el punto tangente $\mathbf{x}=\mathbf{t}$ y teniendo en cuenta que $\nabla D_M^2(\mathbf{x}, \mathbf{m}_j) = 2 \cdot D_M(\mathbf{x}, \mathbf{m}_j) \cdot \nabla D_M(\mathbf{x}, \mathbf{m}_j)$, la condición para el punto de tangencia se puede expresar también como

Gradientes en el pto. tangente

$\nabla p_1(\mathbf{t}) = \beta \cdot \nabla p_2(\mathbf{t})$

$\nabla D_M^2(\mathbf{t}, \mathbf{m}_1) = \alpha \cdot \nabla D_M^2(\mathbf{t}, \mathbf{m}_2)$

$\nabla D_M(\mathbf{t}, \mathbf{m}_1) = \gamma \cdot \nabla D_M(\mathbf{t}, \mathbf{m}_2)$

$\gamma = \alpha \cdot \frac{D_M(\mathbf{t}, \mathbf{m}_2)}{D_M(\mathbf{t}, \mathbf{m}_1)}$

$$\nabla D_M^2(\mathbf{t}, \mathbf{m}_1) = \alpha \cdot \nabla D_M^2(\mathbf{t}, \mathbf{m}_2) , \tag{2.10}$$

siendo α constante real negativa dada por

$$\alpha = \beta \cdot \frac{p_2(\mathbf{t})}{p_1(\mathbf{t})} . \tag{2.11}$$

Aplicando el operador gradiente a $D_M(\mathbf{x}, \mathbf{m}_j)$ (2.5), se obtiene

$$\nabla D_M(\mathbf{x}, \mathbf{m}_j) = \frac{1}{D_M(\mathbf{x}, \mathbf{m}_j)} \cdot \Sigma_j^{-1} (\mathbf{x}-\mathbf{m}_j) , \tag{2.12}$$

por lo que

$$\nabla D_M^2(\mathbf{x}, \mathbf{m}_j) = 2\Sigma_j^{-1} (\mathbf{x}-\mathbf{m}_j) , \tag{2.13}$$

e introduciendo (2.13) en (2.10) y resolviendo para $\mathbf{t}$, se obtiene

EXPRESIÓN ANALÍTICA

Punto de tangencia DTC

$$\mathbf{t}(\alpha) = \left(\Sigma_1^{-1} - \alpha\Sigma_2^{-1}\right)^{-1} \left(\Sigma_1^{-1}\mathbf{m}_1 - \alpha\Sigma_2^{-1}\mathbf{m}_2\right) , \tag{2.14}$$

que representa la expresión general de un punto de tangencia. Por otra parte, para obtener la expresión de los parámetros se sustituye (2.9) en (2.3a) y se particulariza para $\mathbf{x}=\mathbf{t}$,

$$\mathbf{w}_j = -p_j(\mathbf{t}) \cdot D_M(\mathbf{t}, \mathbf{m}_j) \cdot \nabla D_M(\mathbf{t}, \mathbf{m}_j) . \tag{2.15}$$

introduciendo (2.12), resulta

$$\mathbf{w}_j = -p_j(\mathbf{t}) \cdot \Sigma_j^{-1}(\mathbf{t} - \mathbf{m}_j) \ . \tag{2.16}$$

Adviértase que el discriminante resultante es equivalente[1] si se elimina la constante del gradiente $p_j(\mathbf{t})$ de ambos parámetros $\mathbf{w}$ y ω_0.

[1] Dos discriminantes lineales son equivalentes si sus vectores de pesos y sesgos son igualmente proporcionales

Ahora, la regla de decisión para un discriminante lineal basado en DTC se puede escribir como sigue

$$h(\mathbf{x}) = (\mathbf{w}^{\text{DTC}})^T\mathbf{x} + w_0^{\text{DTC}} \underset{C_2}{\overset{C_1}{\gtrless}} 0 \ , \tag{2.17}$$

Regla de decisión DTC

donde

$$\mathbf{w}^{\text{DTC}} = \Sigma_j^{-1}(\mathbf{t}(\alpha) - \mathbf{m}_j) \ , \tag{2.18a}$$

$$\omega_0^{\text{DTC}} = -(\mathbf{w}^{\text{DTC}})^T\mathbf{t}(\alpha) \ , \tag{2.18b}$$

Parámetros DTC

con j=1, 2, y $\mathbf{t}(\alpha)$ dada por (2.14) con $\alpha < 0$. Adviértase que (2.18) son las expresiones particulares de (2.3) cuando los datos están distribuidos normalmente o las clases están descritas por sus dos primeros momentos. También se puede observar que un discriminante lineal descrito por DTC está completamente determinado por el punto tangente que, a su vez, es función del parámetro α.

La expresión explícita de $\mathbf{w}^{\text{DTC}}$ en términos de α se puede obtener introduciendo (2.13) en (2.10), con $\mathbf{x} = \mathbf{t}$, $\mathbf{w}_1 = \alpha\,\mathbf{w}_2$, y eliminando $\mathbf{t}$, obteniendo

$$\mathbf{w}^{\text{DTC}} = \left[\Sigma_1 - \alpha^{-1}\Sigma_2\right]^{-1}(\mathbf{m}_2 - \mathbf{m}_1) \ . \tag{2.19}$$

Parámetros DTC simplificados

Ese resultado permite despejar $\mathbf{t}(\alpha)$ de (2.18a)

$$\mathbf{t}(\alpha) = \Sigma_1\left[\Sigma_1 - \alpha^{-1}\Sigma_2\right]^{-1}(\mathbf{m}_2 - \mathbf{m}_1) + \mathbf{m}_1 \ , \tag{2.20}$$

con la ventaja de que sólo requiere el cálculo de una inversa y no cinco como en (2.14). Esta reducción computacional también se aplica al cálculo de $\mathbf{w}^{\text{DTC}}$ en (2.19) y ω_0^{DTC} en (2.18b). El cálculo de los parámetros del discriminante DTC, dado α, se resume a continuación

Cálculo de los parámetros del discriminante DTC:

1. Calcular el punto de tangencia $\mathbf{t}$ (2.20)
2. Calcular el vector de pesos $\mathbf{w}^{\text{DTC}}$ (2.19)
3. Calcular el sesgo ω_0^{DTC} (2.18b)

2.2 *Relación analítica con el diseño paramétrico*

Una vez que el punto tangente $\mathbf{t}$ se ha expresado en función de α, la expresión de $\mathbf{w}^{\text{DTC}}(\alpha)$ (2.19) es idéntica a la obtenida por el método

paramétrico $\mathbf{w}^{\mathrm{P}}(s)$ (1.5), si

$$s = \frac{\alpha}{\alpha - 1} \tag{2.21}$$

y

$$\mathbf{w}^{\mathrm{DTC}} = \left(\frac{\alpha}{\alpha - 1}\right) \mathbf{w}^{\mathrm{P}} \ . \tag{2.22}$$

MÉTODO DTC
$\mathbf{w}^{\mathrm{DTC}}(\alpha)$ (2.19), $\omega_0^{\mathrm{DTC}}(\alpha)$ (2.18b)

MÉTODO PARAMÉTRICO
$\mathbf{w}^{P}(s)$ (1.5), $\omega_0^{P}(s)$ (1.7)

Es decir, para transformar el $\mathbf{w}^{\mathrm{P}}$ del método paramétrico en el $\mathbf{w}^{\mathrm{DTC}}$ del método DTC o viceversa, primero, es necesario cambiar las variables s y α siguiendo (2.21), y segundo, multiplicar o dividir por $(\frac{\alpha}{\alpha-1})$ respectivamente, como muestra (2.22).

Demostración de (2.22) y (2.21).

Multiplicando (2.19) por $(\frac{\alpha-1}{\alpha})$ ofrece

$$\left(\frac{\alpha - 1}{\alpha}\right) \mathbf{w}^{\mathrm{DTC}} = \left[\frac{\alpha}{\alpha - 1}\Sigma_1 - \frac{1}{\alpha - 1}\Sigma_2\right]^{-1} (\mathbf{m}_2 - \mathbf{m}_1) \ . \tag{2.23}$$

Definiendo

$$s = \frac{\alpha}{\alpha - 1} \ , \tag{2.24}$$

entonces (2.23) resulta

$$\left(\frac{\alpha - 1}{\alpha}\right) \mathbf{w}^{\mathrm{DTC}} = [s\Sigma_1 + (1 - s)\Sigma_2]^{-1} (\mathbf{m}_2 - \mathbf{m}_1) \ , \tag{2.25}$$

donde la parte derecha es igual a $\mathbf{w}^{\mathrm{P}}$ en (1.5) correspondiente al diseño paramétrico de discriminantes lineales. Esto prueba (2.22) con (2.21).

Adviértase que, como se comentó en el Capítulo 1.1, el $\mathbf{w}^{\mathrm{P}}$ (1.5) óptimo está expresado en términos del parámetro s y no depende explícitamente del criterio de separabilidad f.

La expresión analítica de ω_0 depende directamente del criterio de separabilidad f, de este modo cada selección particular de f producirá una expresión particular de ω_0. De esta manera, el valor del sesgo, sustituyendo $\mathbf{t}$ (2.20) en ω_0^{DTC} (2.18b), es

Valor umbral ω_0

$$\omega_0^{\mathrm{DTC}}(\alpha) = -(\mathbf{w}^{\mathrm{DTC}})^T \frac{\Sigma_1 \mathbf{m}_2 - \alpha^{-1}\Sigma_2 \mathbf{m}_1}{\Sigma_1 - \alpha^{-1}\Sigma_2} \ . \tag{2.26}$$

2.2.1 *Discriminante de Bayes para distribuciones Gaussianas*

Además de la equivalencia (2.22) y (2.21) para $\mathbf{w}$ óptimo, el valor del sesgo ω_0^{DTC} (2.18b), con la equivalencia de s (2.21), es el mismo que el obtenido para el discriminante lineal de Bayes por el método paramétrico ω_0^{PB} (1.11) si

$$\omega_0^{\mathrm{DTC}} = \left(\frac{\alpha}{\alpha - 1}\right) \omega_0^{\mathrm{PB}} \ . \tag{2.27}$$

MÉTODO PARAMÉTRICO
DISC. LINEAL BAYES
f: error de Bayes (1.8)
SOLUCIÓN:
$\mathbf{w}^{PB}(s^*)$ (1.5), $\omega_0^{PB}(s^*)$ (1.11)
s^*: óptimo (error de Bayes mínimo)

Por tanto, el discriminante DTC es equivalente al discriminante lineal de Bayes para distribuciones Gaussianas o, según el teorema central del límite, distribuciones con una dimensión n elevada, si se realiza la transformación de los parámetros $\mathbf{w}^{\text{DTC}}$ (2.22) y ω_0^{DTC} (2.27) a partir de los parámetros del discriminante lineal de Bayes. Adviértase que ambos parámetros están multiplicados por el mismo escalar y que por tanto los discriminantes son equivalentes.

Demostración de (2.27).

Introduciendo (2.22) en (2.18b) y considerando que $s = \frac{\alpha}{\alpha-1}$, se obtiene

$$\omega_0^{\text{DTC}} = -s(\mathbf{w}^{\text{P}})^T\mathbf{t} \,. \tag{2.28}$$

De manera similar, introduciendo (2.22) en (2.18a) con $j=1$ y resolviendo para $\mathbf{t}$, resulta

$$\mathbf{t} = s\,\Sigma_1\mathbf{w}^{\text{P}} + \mathbf{m}_1 \,, \tag{2.29}$$

que, introducido en (2.28), produce

$$\omega_0^{\text{DTC}} = -s^2(\mathbf{w}^{\text{P}})^T\Sigma_1\mathbf{w}^{\text{P}} - s(\mathbf{w}^{\text{P}})^T\mathbf{m}_1 \,. \tag{2.30}$$

El objetivo es probar que

$$\omega_0^{\text{P}} = \frac{1}{s}\,\omega_0^{\text{DTC}} \,. \tag{2.31}$$

Introduciendo (1.11) y (2.30) en (2.31), se obtiene

$$-\frac{s\sigma_1^2(\mathbf{w}^{\text{P}})^T\mathbf{m}_2 + (1-s)\sigma_2^2(\mathbf{w}^{\text{P}})^T\mathbf{m}_1}{s\sigma_1^2 + (1-s)\sigma_2^2} = -s(\mathbf{w}^{\text{P}})^T\Sigma_1\mathbf{w}^{\text{P}} - (\mathbf{w}^{\text{P}})^T\mathbf{m}_1 \,. \tag{2.32}$$

Teniendo en cuenta que $\sigma_i^2 = (\mathbf{w}^{\text{P}})^T\Sigma_i\mathbf{w}^{\text{P}}$ (ver (1.4b)), se puede ver fácilmente que (2.32) queda

$$(\mathbf{w}^{\text{P}})^T\{[s\Sigma_1 + (1-s)\Sigma_2]\,\mathbf{w}^{\text{P}} - (\mathbf{m}_2 - \mathbf{m}_1)\} = 0 \,. \tag{2.33}$$

Esta ecuación es siempre verdadera debido a que el término $\{.\}$ es cero (ver (1.5)). Por esta razón, (2.31) es verdadera, y esto prueba (2.27).

2.2.2 *Discriminante de Fisher*

Como se muestra en la Sección 1.1.2, el discriminante lineal de Fisher se obtiene de $\mathbf{w}^{\text{P}}$ (1.5) del método paramétrico con $s = 0.5$, que da como resultado $\mathbf{w}^{\text{PF}}$ (1.14). Según el cambio de variable (2.21), la equivalencia con DTC es $\alpha = -1$, lo que permite obtener $\mathbf{w}^{\text{DTC}}$ a partir de (2.19).

$$\mathbf{w}^{\text{DTC}} = \left(\frac{\alpha}{\alpha-1}\right)\mathbf{w}^{\text{PF}} \,. \tag{2.34}$$

Por otra parte, en contraste con el discriminante clásico de Fisher, Fisher-DTC proporciona un sesgo dado por (2.18b). Sustituyendo $\mathbf{t}$ (2.14) en ω_0^{DTC} (2.18b), o usando ω_0^{DTC} (2.26), y particularizando para

$\alpha = -1$, se obtiene la siguiente expresión

$$\omega_0^{\text{DTC}} = -(\mathbf{w}^{\text{DTC}})^T \frac{\Sigma_2 \mathbf{m}_1 + \Sigma_1 \mathbf{m}_2}{\Sigma_2 + \Sigma_1} \,. \tag{2.35}$$

En este sentido, Fisher-DTC es un discriminante lineal completo y no sólo una dirección de proyección óptima.

2.2.3 *Discriminante basado en Scatter*

Como se muestra en la Sección 1.1.3, el discriminante lineal basado en Scatter se obtiene de $\mathbf{w}^{\text{P}}$ (1.5) con $s = P_1$, que resulta en $\mathbf{w}^{\text{PS}}$ (1.17). De acuerdo con el cambio de variable (2.21), la equivalencia con DTC es $\alpha = -P_1/P_2$, *i.e.*, la relación entre las probabilidades *a priori*, permitiendo obtener la dirección de este discriminante $\mathbf{w}^{\text{DTC}}$ (2.19) a partir de

$$\mathbf{w}^{\text{DTC}} = \left(\frac{\alpha}{\alpha - 1}\right) \mathbf{w}^{\text{PS}} \,. \tag{2.36}$$

Por otra parte, el método DTC proporciona un sesgo por medio de ω_0^{DTC} (2.18b) con $\alpha = -P_1/P_2$. Sustituyendo $\mathbf{t}$ (2.14) en ω_0^{DTC} (2.18b), o directamente usando (2.26), y particularizando para ese valor α, se obtiene

$$\omega_0^{\text{DTC}} = -(\mathbf{w}^{\text{DTC}})^T \frac{P_1 \Sigma_1 \mathbf{m}_2 + P_2 \Sigma_2 \mathbf{m}_1}{P_1 \Sigma_1 + P_2 \Sigma_2} \,, \tag{2.37}$$

que es diferente a la obtenida con el método paramétrico ω_0^{PS} (1.19).

2.3 *Relación analítica con el diseño mediante optimización convexa minimax*

Método directo

El método directo de DTC para el cálculo del punto tangente es una alternativa competitiva en coste computacional a la solución iterativa del método de optimización (SOCP). Está basado en un resultado particular de Clark (1995), donde se estudian estimadores de parámetros multidimensionales que producen estimaciones normales. Proporciona (Teorema 2 de Clark, 1995) condiciones suficientes para que dos elipsoides sean tangentes con interiores disjuntos, y un procedimiento para encontrar el correspondiente punto tangente, presentado en la Sección 2.5.1. Este procedimiento sólo requiere encontrar las raíces de un polinomio de grado $2n$.

La probabilidad de clasificación correcta de los puntos de una clase, usando la distancia de Mahalanobis $D_M(\mathbf{t}^*, \mathbf{m}_j)$ de la media al punto de tangencia $\mathbf{t}^*$, es dada por la función de distribución acumulada

$$\varepsilon_j = \int\limits_{\mathbf{x} \in \mathcal{R}_j} \frac{1}{\sqrt{(2\pi)^n |\Sigma_j|}} \exp\left(-\frac{1}{2} D_M^2(\mathbf{x}, \mathbf{m}_j)\right) d\mathbf{x} \,, \tag{2.38}$$

donde $\mathcal{R}_j = \{\mathbf{x} \in \mathbb{R}^n \mid D_M^2(\mathbf{x}, \mathbf{m}_j) \leq D_M^2(\mathbf{t}, \mathbf{m}_j)\}$ es la región que encierra una probabilidad de acierto ε_j para las muestras de la clase C_j y

está definida por el conjunto de puntos que presentan una distancia de Mahalanobis a la media menor que la distancia de Mahalanobis de la media al punto tangente $\mathbf{t}$. La probabilidad de acierto en función de la distancia de Mahalanobis de la media al punto tangente se usa para establecer la correspondencia de los discriminantes DTC con el método de optimización convexa minimax de la Sección 1.2, como se muestra a continuación.

2.3.1 *Discriminante de Bayes para distribuciones Gaussianas*

El discriminante lineal de Bayes se obtiene minimizando el error de Bayes en el espacio proyectado por el vector de dirección del discriminante. A partir de la regla de Bayes

$$P_1 \frac{1}{\sqrt{(2\pi)^n|\Sigma_1|}} \exp\left(-\frac{1}{2}D_M^2(\mathbf{x},\mathbf{m}_1)\right) = P_2 \frac{1}{\sqrt{(2\pi)^n|\Sigma_2|}} \exp\left(-\frac{1}{2}D_M^2(\mathbf{x},\mathbf{m}_2)\right), \tag{2.39}$$

siendo P_j la probabilidad *a priori* de la clase C_j, $j=1,2$, y D_M la distancia de Mahalanobis dada por (2.5), la optimización procede de su integración (2.40) en ambos lados para tener una expresión dependiente de la función de distribución acumulada, que es la probabilidad de acierto de cada clase ε

$$\underbrace{P_1}_{\lambda_1}\underbrace{\int_{\mathbf{x}\in\mathcal{R}_1} \frac{1}{\sqrt{(2\pi)^n|\Sigma_1|}} \exp\left(-\frac{1}{2}D_M^2(\mathbf{x},\mathbf{m}_1)\right)d\mathbf{x}}_{\varepsilon_1} + \underbrace{P_2}_{\lambda_2}\underbrace{\int_{\mathbf{x}\in\mathcal{R}_2} \frac{1}{\sqrt{(2\pi)^n|\Sigma_2|}} \exp\left(-\frac{1}{2}D_M^2(\mathbf{x},\mathbf{m}_2)\right)d\mathbf{x}}_{\varepsilon_2}, \tag{2.40}$$

obteniéndose una expresión, cuya maximización es equivalente a la del problema de optimización (1.21).

2.3.2 *Discriminante MPDH-DTC*

La solución MPDH tiene como objetivo obtener el hiperplano que separa las clases con máxima e igual probabilidad de éxito (Lanckriet et al., 2002). Geométricamente, el punto de tangencia de la frontera del discriminante con ambos elipsoides de igual nivel de probabilidad se encuentra en la equidistancia de Mahalanobis a las medias, el punto que hace mínimo el máximo de su distancia a la media de cada clase; *i.e.*, dado un punto arbitrario, se calculan las distancias a cada una de las medias de las clases y la mayor de ellas se minimiza (si durante el proceso la mayor distancia es la distancia a la otra media, se minimiza esta distancia).

La relación entre las distancias de Mahalanobis desde el punto tangente $\mathbf{t}(\alpha)$ a cada media $\mathbf{m}_j$, $j=1,2$, satisface

$$\frac{D_M(\mathbf{t},\mathbf{m}_1)}{D_M(\mathbf{t},\mathbf{m}_2)} = \frac{\sqrt{(\mathbf{t}-\mathbf{m}_1)^T\Sigma_1^{-1}(\mathbf{t}-\mathbf{m}_1)}}{\sqrt{(\mathbf{t}-\mathbf{m}_2)^T\Sigma_2^{-1}(\mathbf{t}-\mathbf{m}_2)}} = \frac{r_1}{r_2}. \tag{2.41}$$

DTC

Si $r_1 = r_2$, el DTC descrito por (2.41) determina un punto tangente $\mathbf{t}^*$ con máxima e igual distancia de Mahalanobis a $\mathbf{m}_1$ y $\mathbf{m}_2$,

$$D_M(\mathbf{t}^*, \mathbf{m}_1) - D_M(\mathbf{t}^*, \mathbf{m}_2) = 0 \,. \tag{2.42}$$

Por lo tanto, MPDH es dado aquí por un DTC particular llamado discriminante MPDH-DTC, que es una alternativa no iterativa a los métodos de optimización empleados en MPM.

Método de optimización minimax

Como se expuso anteriormente, la equidistancia de Mahalanobis desde el punto tangente a ambas medias implica que el discriminante separa ambas clases con igual probabilidad. Así, la probabilidad de acierto encerrada por una distancia de Mahalanobis $D_M(\mathbf{t}^*, \mathbf{m}_j)$ es dada por la función de distribución acumulada (2.38). Considerando $D_M(\mathbf{t}^*, \mathbf{m}_1) = D_M(\mathbf{t}^*, \mathbf{m}_2)$, la probabilidad de éxito de cada clase es también igual, verificándose

$$\underbrace{\int_{\mathbf{x}\in\mathcal{R}_1} \frac{1}{\sqrt{(2\pi)^n |\Sigma_1|}} \exp\left(-\frac{1}{2} D_M^2(\mathbf{x}, \mathbf{m}_1)\right) d\mathbf{x}}_{\varepsilon_1} = \underbrace{\int_{\mathbf{x}\in\mathcal{R}_2} \frac{1}{\sqrt{(2\pi)^n |\Sigma_2|}} \exp\left(-\frac{1}{2} D_M^2(\mathbf{x}, \mathbf{m}_2)\right) d\mathbf{x}}_{\varepsilon_2} \,. \tag{2.43}$$

De esta forma, si $\varepsilon_1 = \varepsilon_2 = \varepsilon$ y el objetivo es maximizar la probabilidad de éxito ε, existe una equivalencia con la expresión del problema de optimización (1.22).

2.4 *Discriminante Quasi-Bayes-DTC*

En esta sección se presenta un nuevo discriminante DTC que por sus características de diseño se ha denominado Quasi-Bayes-DTC. El punto de tangencia DTC que define el discriminante es el punto de cruce entre la frontera óptima de Bayes para distribuciones Gaussianas y la curva de todos los puntos DTC, Figura 2.4. Como se verá en la sección de experimentos, produce resultados de precisión muy cercanos a los del discriminante lineal de Bayes para distribuciones Gaussianas con un coste computacional menor debido a que es una solución directa (no se requiere la búsqueda de ningún parámetro). Por último, se demostrará la relación de este discriminante con los discriminantes obtenidos mediante optimización convexa minimax.

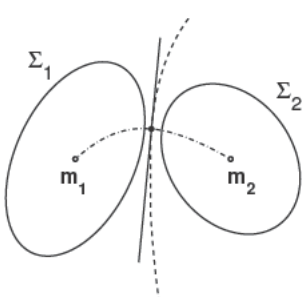


Figura 2.4: Discriminante lineal Quasi-Bayes DTC. Este discriminante (línea en negrita) está descrito por el punto tangente DTC dado por la intersección entre la frontera cuadrática de Bayes (curva a trazos) y la curva de todos los puntos de tangencia posibles de DTC (curva a trazos y puntos). El punto de tangencia Quasi-Bayes DTC se muestra con un punto en negrita.

El cálculo del discriminante se hace de manera similar a MPDH-DTC, pero la equidistancia de Mahalanobis se reemplaza por una diferencia de distancias de Mahalanobis no nula. El punto de tangencia $\mathbf{t}^*$ de Quasi-Bayes-DTC, ya que también es el punto de tangencia de la frontera cuadrática de Bayes, satisface la regla de Bayes (2.39).

De manera similar a (2.42), tomando logaritmos en (2.39)

$$D_M^2(\mathbf{t}^*, \mathbf{m}_1) - D_M^2(\mathbf{t}^*, \mathbf{m}_2) = K \,, \tag{2.44}$$

donde

$$K = \ln\left[\left(\frac{P_1}{P_2}\right)^2 \frac{|\Sigma_2|}{|\Sigma_1|}\right] \,. \tag{2.45}$$

Este discriminante proporciona una buena clasificación en términos de probabilidad de error cuando las probabilidades *a priori* son una parte importante del problema –como se ha visto en problemas desequilibrados (*imbalanced*) con valores extremos de las probabilidades *a priori*– ya que siempre se mueve con el discriminante no lineal de Bayes óptimo.

Método de optimización minimax

Para establecer la relación con el método de optimización minimax, sea $D'^2_M(\mathbf{t}^*, \mathbf{m}_2) = D^2_M(\mathbf{t}^*, \mathbf{m}_2) + K$ una nueva distancia de Mahalanobis desde la media de la clase C_2 que es igual a la distancia de Mahalanobis desde la media de la clase C_1 al punto tangente del discriminante

$$D^2_M(\mathbf{t}^*, \mathbf{m}_1) = D'^2_M(\mathbf{t}^*, \mathbf{m}_2) \; . \tag{2.46}$$

La probabilidad de éxito encerrada por estas distancias de Mahalanobis es

$$\underbrace{\int\limits_{\mathbf{x}\in\mathcal{R}_1} \frac{1}{\sqrt{(2\pi)^n|\Sigma_1|}} \exp\left(-\frac{1}{2}D^2_M(\mathbf{x},\mathbf{m}_1)\right)d\mathbf{x}}_{\varepsilon_1} = \underbrace{\int\limits_{\mathbf{x}\in\mathcal{R}_2} \frac{1}{\sqrt{(2\pi)^n|\Sigma_2|}} \exp\left(-\frac{1}{2}D'^2_M(\mathbf{x},\mathbf{m}_2)\right)d\mathbf{x}}_{\varepsilon'_2} , \tag{2.47}$$

donde

$$\varepsilon'_2 = \int\limits_{\mathbf{x}\in\mathcal{R}_2} \frac{1}{\sqrt{(2\pi)^n|\Sigma_2|}} \exp\left(-\frac{1}{2}\left(D^2_M(\mathbf{x},\mathbf{m}_2)+K\right)\right)d\mathbf{x} = \Gamma\cdot\varepsilon_2, \tag{2.48}$$

siendo ε_2 descrita por (2.38) y resultando $\varepsilon_1 = \Gamma\cdot\varepsilon_2$, con

$$\Gamma = \frac{P_2}{P_1}\sqrt{\frac{|\Sigma_1|}{|\Sigma_2|}} \; . \tag{2.49}$$

Así, el problema de optimización, siguiendo su expresión general (1.20), y usando los resultados obtenidos en MPDH (2.43)-(1.22) es

QUASI-BAYES (método de optimización minimax)

$$\begin{aligned} \max_{\varepsilon,\mathbf{w}\neq\mathbf{0},\omega_0} \; \varepsilon \quad s.t. \quad & \inf_{\mathbf{x}\sim(\mathbf{m}_1,\Sigma_1)} P\{\mathbf{w}^T\mathbf{x}+\omega_0 \geq 0\} \geq \varepsilon \\ & \inf_{\mathbf{x}\sim(\mathbf{m}_2,\Sigma_2)} P\{\mathbf{w}^T\mathbf{x}+\omega_0 \leq 0\} \geq \Gamma\cdot\varepsilon \; , \end{aligned} \tag{2.50}$$

2.5 *Cálculo de los discriminantes DTC*

El procedimiento del cálculo del discriminante DTC se resume a continuación. Primero, se estiman los dos primeros momentos de las distribuciones de las clases. En segundo lugar, se calcula el parámetro α, que determina un único discriminante DTC. Este cálculo puede provenir del cálculo del parámetro s por medio del método paramétrico para luego establecer la correspondencia con el α del método DTC; o, en el caso de MPDH y Quasi-Bayes, el valor α se calcula a través del polinomio de Clark, como se muestra en la Sección 2.5.1.

Una vez que se conoce α, el cálculo del punto tangente $\mathbf{t}$ es directo, y posteriormente los parámetros $\mathbf{w}^{\text{DTC}}$ y ω_0^{DTC}, que determinan el discriminante DTC. La Sección 2.5.2 presenta más exhaustivamente el algoritmo DTC

Cálculo de los discriminantes DTC:

1. Estimación de $\mathbf{m}_1$, $\mathbf{m}_2$, Σ_1 y Σ_2 a partir de los datos.
2. Determinar α^*:
   - A partir del método paramétrico
     - Aplicando la transformación $s \to \alpha$ (2.22)
   - A través del polinomio de Clark
     - Problema MPDH con $K=0$
     - Discriminante Quasi-Bayes con K de (2.45)
3. Calcular el punto de tangencia $\mathbf{t}(\alpha^*)$ (2.20)
4. Calcular el vector de pesos $\mathbf{w}^{\text{DTC}}$ (2.19) ó (2.18a)
5. Calcular el sesgo ω_0^{DTC} (2.18b)

2.5.1 *Cálculo de α con el polinomio de Clark*

Sean $\mathbf{m}_j \in \mathbb{R}^n$ los vectores media y $\Sigma_j \in \mathbb{R}^{n\times n}$, j=1, 2, las matrices de covarianza de un problema dado de clasificación binaria. El objetivo es encontrar el punto de tangencia DTC. Primero, los datos se trasladan espacialmente de tal manera que una de las medias, *e.g.*, $\mathbf{m}_2$, sea $\mathbf{0}$. Esto se hace restando $\mathbf{m}_2$ a todos los puntos de los datos. Después, la matriz $T \in \mathbb{R}^{n\times n}$ diagonaliza simultáneamente Σ_1^{-1} y Σ_2^{-1} para transformarlas en una matriz diagonal definida positiva, $D \in \mathbb{R}^{n\times n}$, y la matriz identidad $I \in \mathbb{R}^{n\times n}$

$$T^T \Sigma_1^{-1} T = I \,, \tag{2.51a}$$
$$T^T \Sigma_2^{-1} T = D \,. \tag{2.51b}$$

Ahora, cada punto $\mathbf{x}$ de los datos ha sido transformado en $\tilde{\mathbf{x}}$ de acuerdo a

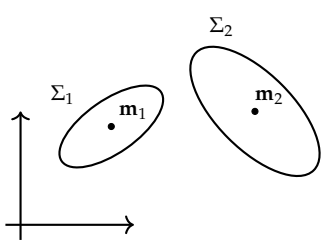
$$\tilde{\mathbf{x}} = T^{-1}(\mathbf{x} - \mathbf{m}_2) \,, \tag{2.52}$$

y las medias transformadas son

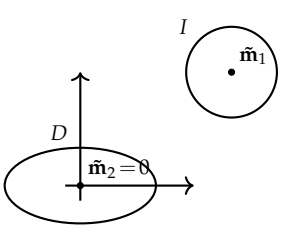
$$\tilde{\mathbf{m}}_1 = T^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \,, \tag{2.53a}$$
$$\tilde{\mathbf{m}}_2 = T^{-1}(\mathbf{m}_2 - \mathbf{m}_2) = 0 \,. \tag{2.53b}$$

Entonces, el nuevo problema es encontrar el punto tangente entre una hiperesfera centrada en $\tilde{\mathbf{m}}_1$ y un elipsoide centrado en el origen $\tilde{\mathbf{m}}_2 = 0$

El Teorema 2 de Clark (1995) proporciona condiciones suficientes para que dos elipsoides sean tangentes con interiores disjuntos y



(a) Espacio original



(b) Espacio transformado

Figura 2.5: Diagonalización simultánea

un procedimiento para encontrar el correspondiente punto tangente. Estas condiciones suficientes son

$$\tilde{\mathbf{t}}(\alpha) = (I - \alpha D)^{-1}\,\tilde{\mathbf{m}}_1\,, \tag{2.54a}$$
$$r_1^2 = (\tilde{\mathbf{t}}(\alpha) - \tilde{\mathbf{m}}_1)^T(\tilde{\mathbf{t}}(\alpha) - \tilde{\mathbf{m}}_1)\,, \tag{2.54b}$$
$$r_2^2 = \tilde{\mathbf{t}}(\alpha)^T D\,\tilde{\mathbf{t}}(\alpha)\,, \tag{2.54c}$$
$$\alpha < 0\,, \tag{2.54d}$$

donde $\tilde{\mathbf{t}}$ es el punto tangente DTC en el espacio transformado. El valor único α^* que satisface las condiciones es la solución real y negativa de

$$G(\alpha) - H(\alpha) = K\,, \tag{2.55}$$

donde

$$K \;= \tilde{D}_M^2(\tilde{\mathbf{t}}, \tilde{\mathbf{m}}_1) - \tilde{D}_M^2(\tilde{\mathbf{t}}, \tilde{\mathbf{m}}_2) \tag{2.56}$$

es la diferencia de las distancias de Mahalanobis del punto tangente a las medias y

$$G(\alpha) = \alpha^2\,\tilde{\mathbf{m}}_1^T\,D^2\,(I - \alpha D)^{-2}\,\tilde{\mathbf{m}}_1\,, \tag{2.57a}$$
$$H(\alpha) = \tilde{\mathbf{m}}_1^T\,D\,(I - \alpha D)^{-2}\,\tilde{\mathbf{m}}_1\,. \tag{2.57b}$$

Para encontrar la solución, $G(\alpha)$ y $H(\alpha)$ se expresan como

$$G(\alpha) = \frac{\alpha^2\,C(\alpha)}{A(\alpha)}\,, \qquad H(\alpha) = \frac{B(\alpha)}{A(\alpha)} \tag{2.58}$$

donde $A(\alpha)$ es un polinomio de grado $2n$, y $B(\alpha)$ y $C(\alpha)$ son polinomios de grado $2n-2$ o menor. Usando este resultado, la solución de (2.55) es la única raíz real del polinomio de Clark de grado $2n$

$$\alpha^2\,C(\alpha) - B(\alpha) - KA(\alpha) = 0\,, \tag{2.59}$$

que satisface $\alpha < 0$. $A(\alpha)$, $B(\alpha)$ y $C(\alpha)$ se puede obtener computacionalmente usando el Algoritmo 1 de Clark (1995).

Una vez obtenida la solución α^*, el punto tangente DTC $\tilde{\mathbf{t}}(\alpha^*)$ en el espacio transformado se calcula con (2.54a), y el correspondiente punto tangente $\mathbf{t}$ en el espacio original se puede obtener de (2.52) con $\mathbf{x} = \mathbf{t}(\alpha^*)$, $\tilde{\mathbf{x}} = \tilde{\mathbf{t}}(\alpha^*)$, y despejando $\mathbf{t}$

$$\mathbf{t}(\alpha^*) = T\,\tilde{\mathbf{t}}(\alpha^*) + \mathbf{m}_2\,. \tag{2.60}$$

Este resultado se puede obtener también de (2.14) ó (2.20) con α^*. Esto es cierto porque se puede obtener (2.54) del análisis DTC.

Demostración. Las condiciones de Clark (2.54) se pueden obtener del análisis DTC en el espacio transformado

Considerando las distancias de Mahalanobis en el espacio transformado del punto tangente $\tilde{\mathbf{t}}$ a las medias $\tilde{\mathbf{m}}_j$, j=1, 2, ($\tilde{\mathbf{m}}_2 = 0$),

$$\tilde{D}_M(\tilde{\mathbf{t}}, \tilde{\mathbf{m}}_j) = r_j\,, \tag{2.61}$$

adviértase que las condiciones (2.54b) y (2.54c) son las siguientes distancias de Mahalanobis particularizadas en $\tilde{\mathbf{x}}=\tilde{\mathbf{t}}$

$$\tilde{D}^2_M(\tilde{\mathbf{x}},\tilde{\mathbf{m}}_1) = (\tilde{\mathbf{x}}-\tilde{\mathbf{m}}_1)^T(\tilde{\mathbf{x}}-\tilde{\mathbf{m}}_1)\,, \tag{2.62a}$$

$$\tilde{D}^2_M(\tilde{\mathbf{x}},\mathbf{0}) = \tilde{\mathbf{x}}^T D\,\tilde{\mathbf{x}}\,, \tag{2.62b}$$

y aplicando el operador gradiente y particularizando en $\tilde{\mathbf{x}}=\tilde{\mathbf{t}}$

$$\nabla\tilde{D}^2_M(\tilde{\mathbf{t}},\tilde{\mathbf{m}}_1) = 2(\tilde{\mathbf{t}}-\tilde{\mathbf{m}}_1)\,, \tag{2.63a}$$

$$\nabla\tilde{D}^2_M(\tilde{\mathbf{t}},\mathbf{0}) = 2\,D\,\tilde{\mathbf{t}}\,. \tag{2.63b}$$

Usando (2.10) del análisis DTC

$$\nabla\tilde{D}^2_M(\tilde{\mathbf{t}}(\alpha),\tilde{\mathbf{m}}_1) = \alpha\cdot\nabla\tilde{D}^2_M(\tilde{\mathbf{t}}(\alpha),\mathbf{0})\,. \tag{2.64}$$

Sustituyendo (2.63) en (2.64)

$$2(\tilde{\mathbf{t}}-\tilde{\mathbf{m}}_1) = 2\,\alpha D\,\tilde{\mathbf{t}}\,, \tag{2.65}$$

y despejando $\tilde{\mathbf{t}}$

$$\tilde{\mathbf{t}} = (I-\alpha D)^{-1}\,\tilde{\mathbf{m}}_1\,, \tag{2.66}$$

que es igual que la condición (2.54a), probando que las condiciones de Clark se pueden obtener del análisis DTC.

Adviértase que la solución α^* del polinomio de Clark (2.59) es la solución del problema MPDH si $K=0$ (2.56), debido a que la diferencia de distancias de Mahalanobis en el punto tangente es nula, $r_1=r_2=r$. Por otra parte, la solución del polinomio de Clark es el discriminante Quasi-Bayes si la diferencia de las distancias de Mahalanobis K en el punto tangente satisface la regla de Bayes, como se calcula en (2.45).

2.5.2 *Algoritmo DTC*

Este método para determinar el discriminante DTC sigue el cálculo del punto tangente de la Sección 2.5.1 a través del polinomio de Clark.

La entrada del algoritmo son los patrones etiquetados y la salida son los parámetros $\mathbf{w}^*$ y ω_0^* del discriminante DTC.

Primero, se estiman los dos primeros momentos de las clases y las probabilidades a priori: $\mathbf{m}_j$, Σ_j, y P_j, $j=1,2$. Entonces, se calcula la matriz de transformación T (Sección 2.5.1) usando la factorización de Cholesky y la descomposición de Schur para diagonalizar simultáneamente ambas matrices inversas de covarianza. El resultado de la factorización de Cholesky de la matriz de covarianza Σ_1 es la matriz G, que satisface

$$GG^T = \Sigma_1^{-1}\,. \tag{2.67}$$

Usando

$$F = G^{-1}\Sigma_2^{-1}(G^T)^{-1}\,, \tag{2.68}$$

la descomposición de Schur de F resulta en Q y Z, con Z matriz

diagonal, que satisfacen

$$Q^T F Q = Z \,. \tag{2.69}$$

Entonces, es posible calcular la matriz de transformación T como

$$T = (G^T)^{-1} Q \,. \tag{2.70}$$

La matriz diagonal y la matriz identidad después de la diagonalización simultanea son

$$D = T^T \Sigma_2^{-1} T \,, \tag{2.71a}$$
$$I = T^T \Sigma_1^{-1} T \,, \tag{2.71b}$$

siendo D una matriz diagonal definida positiva. El Algoritmo 1 de Clark (1995) permite construir el polinomio en α para obtener el punto tangente DTC. Primero, los vectores auxiliares b, la diagonal de la matriz diagonal D, e y, la media no nula en el espacio transformado, se calculan ,

$$\mathbf{b} = \text{diag}(D) \,, \tag{2.72a}$$
$$\mathbf{y} = T^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \,. \tag{2.72b}$$

A partir de aquí, los subíndices numéricos se refieren a las posiciones de un vector, excepto $\mathbf{x}_j$ y Σ_j donde los subíndices todavía se refieren a las clases, j=1,2. b e y se usan para calcular los coeficientes del polinomio en α

$$C_1(\alpha) = \mathbf{b}_1^2 \mathbf{y}_1^2 \,, \tag{2.73a}$$
$$B_1(\alpha) = \mathbf{b}_1 \mathbf{y}_1^2 \,, \tag{2.73b}$$
$$A_1(\alpha) = \mathbf{b}_1^2 \alpha^2 - 2\mathbf{b}_1 \alpha + 1 \,, \tag{2.73c}$$

para calcular iterativamente, de k=2 a n, siendo n la dimensión de los datos,

$$\begin{aligned} C_k(\alpha) &= \mathbf{b}_k^2 \alpha^2 C_{k-1}(\alpha) - 2\mathbf{b}_k \alpha C_{k-1}(\alpha) \\ &\quad + C_{k-1}(\alpha) + \mathbf{b}_k^2 \mathbf{y}_k^2 A_{k-1}(\alpha) \,, \end{aligned} \tag{2.74a}$$
$$\begin{aligned} B_k(\alpha) &= \mathbf{b}_k^2 \alpha^2 B_{k-1}(\alpha) - 2\mathbf{b}_k \alpha B_{k-1}(\alpha) \\ &\quad + B_{k-1}(\alpha) + \mathbf{b}_k \mathbf{y}_k^2 A_{k-1}(\alpha) \,, \end{aligned} \tag{2.74b}$$
$$\begin{aligned} A_k(\alpha) &= \mathbf{b}_k^2 \alpha^2 A_{k-1}(\alpha) - 2\mathbf{b}_k \alpha A_{k-1}(\alpha) \\ &\quad + A_{k-1}(\alpha) \,. \end{aligned} \tag{2.74c}$$

Posteriormente, se resuelve el polinomio de Clark de grado $2n$ (2.59) en α

$$\alpha^2 C(\alpha) - B(\alpha) - KA(\alpha) = 0 \,, \tag{2.75}$$

para obtener un único α^* real negativo, que es la solución del punto tangente DTC $\mathbf{t}^*$ dado por (2.14)

$$\mathbf{t}^* = \left(\Sigma_1^{-1} - \alpha^* \Sigma_2^{-1}\right)^{-1} \left(\Sigma_1^{-1}\mathbf{m}_1 - \alpha^* \Sigma_2^{-1}\mathbf{m}_2\right) \,. \tag{2.76}$$

Usando el punto tangente $\mathbf{t}^*$ en (2.18)

$$\mathbf{w}^* = \Sigma_1^{-1}(\mathbf{t}^* - \mathbf{m}_1)\ , \tag{2.77a}$$

$$\omega_0^* = -(\mathbf{w}^*)^T\mathbf{t}^*\ , \tag{2.77b}$$

se obtienen los parámetros $\mathbf{w}^*$ y ω_0^* del discriminante lineal DTC.

Adviértase que la solución α^* del polinomio de Clark (2.59) es la solución del problema MPDH si $K = 0$ (2.56). Por otra parte, la solución del polinomio de Clark es el discriminante Quasi-Bayes si la diferencia de las distancias de Mahalanobis K en el punto tangente satisface la regla de Bayes, como se calcula en (2.45).

A continuación, se presenta el algoritmo DTC. Adviértase que A, B, C y P son vectores y n es la dimensión de los datos

Algoritmo 2: Algoritmo DTC

Datos: Patrones etiquetados
Resultado: $\mathbf{w}^*$, ω_0^*
Estimación de $\mathbf{m}_j$, Σ_j y P_j, j=1,2
$G = \text{cholesky}(\Sigma_1^{-1})$
$F = G^{-1}\Sigma_2^{-1}(G^T)^{-1}$
$Q, Z = \text{schur}(F)$
$T = (G^T)^{-1}Q$
$D = T^T\Sigma_2^{-1}T$
$\mathbf{b} = \text{diag}(D)$
$\mathbf{y} = T^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$
$C = \mathbf{b}_1^2\mathbf{y}_1^2$
$B = \mathbf{b}_1\mathbf{y}_1^2$
$A = [\mathbf{b}_1^2,\ -2\mathbf{b}_1,\ 1]$
for k=2 *hasta* n **do**
  $C = \mathbf{b}_k^2 \cdot [C,\ 0,\ 0] - 2\mathbf{b}_k \cdot [0,\ C,\ 0] + [0,\ 0,\ C] + \mathbf{b}_k^2\mathbf{y}_k^2 \cdot A$
  $B = \mathbf{b}_k^2 \cdot [B,\ 0,\ 0] - 2\mathbf{b}_k \cdot [0,\ B,\ 0] + [0,\ 0,\ B] + \mathbf{b}_k\mathbf{y}_k^2 \cdot A$
  $A = \mathbf{b}_k^2 \cdot [A,\ 0,\ 0] - 2\mathbf{b}_k \cdot [0,\ A,\ 0] + [0,\ 0,\ A]$
if *MPDH-DTC* **then**
  $K = 0$
if *Quasi-Bayes-DTC* **then**
  $K = -2\log\left(\frac{P_1}{(1-P_1)}\sqrt{\frac{|\Sigma_2|}{|\Sigma_1|}}\right)$
$P = [C,\ 0,\ 0] - [0,\ 0,\ B] + K \cdot A$
$\alpha^* = \text{roots}(P)$
$\mathbf{t}^* = \left(\Sigma_1^{-1} - \alpha^*\Sigma_2^{-1}\right)^{-1} * \left(\Sigma_1^{-1}\mathbf{m}_1 - \alpha^*\Sigma_2^{-1}\mathbf{m}_2\right)$
$\mathbf{w}^* = \Sigma_1^{-1}(\mathbf{t}^* - \mathbf{m}_1)$
$\omega_0^* = -(\mathbf{w}^*)^T\mathbf{t}^*$

Parte II

Discriminantes DTC no lineales

En algunos casos, se usan los discriminantes lineales para resolver problemas de clasificación debido a su simplicidad. Sin embargo, los clasificadores lineales son a menudo insuficientes para resolver efectivamente otros problemas, por lo que se necesitan arquitecturas no lineales más potentes.

Se puede construir un discriminante no lineal transformando los datos de entrada en un **espacio intermedio** y después usar un discriminante lineal ya que se ha demostrado que, a través de una transformación apropiada, el nuevo conjunto de datos en el espacio transformado tiene una mayor separabilidad lineal (Vapnik, 1995; Huang et al., 2006).

Se puede realizar esta transformación con métodos directos como el Análisis de Componentes Principales (*Principal Components Analysis*, PCA) de Jollife (2002), paradigma de la reducción de dimensionalidad, o por medio de estructuras neuronales más o menos complejas. Entre ellas, las redes alimentadas hacia adelante de una sola capa oculta (*Single-hidden Layer Feedforward Networks*, SLFNs) (Huang et al., 2006; Widrow et al., 2013; Martínez-García y Sancho-Gómez, 2018), que desempeñan la transformación a través de una capa oculta con unidades no lineales, permitiendo resolver el problema de clasificación en la capa de salida con un discriminante lineal. Las SLFNs más representativas son: 1) el Perceptrón Multicapa (*Multi-layer Perceptron*, MLP) con una capa oculta de unidades sigmoidales o unidades lineales rectificadas (*Rectified Linear Units*, ReLU). Estas redes se entrenan usando normalmente algoritmos de gradiente (retropropagación o *Back-propagation*) y se diseñan usando técnicas de validación cruzada (*Cross-Validation*, CV). Presentan el inconveniente del entrenamiento estancado (tanto por caer en mínimos locales, como por el efecto conocido por parálisis) y un diseño computacional costoso;

2) Las redes de funciones de base radial (*Radial Basis Function Networks*, **RBFN**) con una capa oculta de núcleos (*kernels*) Gaussianos. Se usan técnicas de cuantificación vectorial (*vector quantization*) para el cálculo de los núcleos y el algoritmo de mínimos cuadrados (*Least Mean Squares*, LMS) para los pesos de la capa de salida. El diseño del tamaño de la capa oculta se realiza mediante CV. Hay otras vías alternativas de diseñar y entrenar una red RBF pero la descrita destaca por su efectividad (Haykin, 2009); por último, 3) redes de núcleos (*kernel-based networks*) que usan el llamado truco del núcleo (*kernel trick*) para desempeñar la transformación no lineal con el objetivo de diseñar y entrenar la red completa mediante la resolución de un problema de optimización convexa (Schölkopf y Smola, 2001). Éstas son, entre otras, las máquinas de vectores de soporte (*Support Vector Machines*, SVMs). La principal desventaja de este método es su alto coste computacional para conjuntos de muestras grandes.

También existen redes profundas (*deep networks*) que realizan una transformación del espacio de datos que mejora el desempeño obtenido en el espacio original. Entre otras, destacan los autocodificadores apilados de eliminación de ruido (*Stacked Denoising Auto-Encoders*, SDAE) (Vincent et al., 2010), una técnica usada muy a menudo debido a su alta capacidad de representación; las redes generativas adversarias (*Generative Adversarial Nets*, GAN) (Goodfellow et al., 2014), y más concretamente los autocodificadores adversarios (*Adversarial Auto-Encoders*, AAE) (Makhzani et al., 2015); y los autocodificadores variacionales (*Variational Auto-Encoders*, VAE) (Doersch, 2016), en los que el espacio intermedio puede ser usado para aplicar un discriminante lineal.

3

Redes de Funciones de Base Radial (RBFN)

Proyección no lineal

En este capítulo, el espacio original de los datos de entrada $\mathbf{x} \in \mathbb{R}^n$ del problema de clasificación se transforma en un espacio de características de dimensión superior a través de la función de transformación $\phi : \mathbb{R}^n \to \mathbb{R}^m$, tal que la frontera de decisión lineal $\mathcal{H}_{\text{L}}(\mathbf{w}, \omega_0) = \{\phi(\mathbf{x}) \in \mathbb{R}^m \mid \mathbf{w}^T\phi(\mathbf{x}) + \omega_0 = 0\}$ en el espacio proyectado de características $\mathbb{R}^m$ corresponde a la frontera de decisión no lineal en el espacio de datos original $\mathcal{H}_{\text{NL}}(\mathbf{w}, \omega_0) = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{w}^T\phi(\mathbf{x}) + \omega_0 = 0\}$.

Es bien sabido que una proyección no lineal apropiada en un espacio de características intermedio incrementa la separabilidad lineal de las clases, permitiendo el uso de un discriminante lineal (Vapnik, 1995; Huang et al., 2006). Una red alimentada hacia adelante de una sola capa oculta (*Single-hidden Layer Feedforward Neural Network*, SLFN) (Huang et al., 2006; Widrow et al., 2013; Martínez-García y Sancho-Gómez, 2018) es una arquitectura de red neuronal basada en esta idea para construir un discriminante no lineal, en el que los datos de entrada se proyectan por medio de la capa oculta y son clasificados mediante el discriminante lineal de la capa de salida. Por lo tanto, el comportamiento no lineal es proporcionado por la capa oculta a través de los distintos tipos de núcleos. Esta red transforma las N muestras de entrada en $\mathbb{R}^n$ en N muestras en $\mathbb{R}^m$, siendo m el número de nodos ocultos. El número de clases determina la dimensión de la capa de salida, que es, a su vez, el número de nodos de salida o discriminantes lineales. Sin embargo, adviértase que en problemas de clasificación binaria la salida es unidimensional y, por tanto, sólo se necesita un discriminante lineal en la capa de salida.

SLFN

Entrenamiento

A partir de un discriminante lineal particular, es posible construir un clasificador no lineal usando una arquitectura SLFN. Para ello, se requiere el entrenamiento de los pesos de la capa oculta (la parte no lineal) y los de la capa de salida (la parte lineal). Mientras que los MLP y las SVM emplean algoritmos de entrenamiento para toda la estructura –con un buen rendimiento pero con el problema del

atascamiento en mínimos locales en el caso del entrenamiento por descenso de gradiente (retropropagación o *Back-propagation*)– otros SLFNs permiten un entrenamiento independiente de cada capa (Haykin, 2009; Duda et al., 2000) con un entrenamiento no supervisado, *i.e.*, sin tener en cuenta los *targets*, de la capa oculta y supervisado de la capa de salida. El entrenamiento no supervisado más común de los pesos de la capa oculta es su ajuste aleatorio al inicio, como ocurre en ELM (*Extreme Learning Machine*) (Huang et al., 2006) y en el algoritmo *No-Propagation* (Widrow et al., 2013; Martínez-García y Sancho-Gómez, 2018), que incrementa la velocidad de entrenamiento aunque necesita a veces una capa oculta de mayor tamaño. Las técnicas de cuantificación vectorial (*vector quantization*) para transformaciones Gaussianas presentan un entrenamiento no supervisado más preciso, como se verá más adelante. Una vez entrenada la capa oculta, el discriminante lineal de la capa de salida se puede entrenar a través de la pseudoinversa, *e.g.*, ELM; con un algoritmo de gradiente, *e.g.*, los mínimos cuadrados (LMS) de *No-Propagation*, menos sensible al ruido que la pseudoinversa (Martínez-García y Sancho-Gómez, 2018); o un algoritmo paramétrico, *e.g.*, los discriminantes DTC.

No supervisado (capa oculta no lineal)

- Pesos aleatorios
- Cuantificación vectorial

Supervisado (capa de salida lineal)

- Pseudoinversa
- Descenso por gradiente
- Métodos paramétricos

KERNELS

Para realizar el mapeado al espacio de características, existen métodos basados en la memoria en los que las muestras de entrenamiento, o un subconjunto de ellas, se usan para predecir en la fase de test (Bishop, 2006). Consisten en combinaciones lineales de una función de kernel que incluye una medida de similitud de dos muestras en el espacio de entrada. Este método es rápido en el entrenamiento pero lento en test. Para las funciones de mapeo no lineal $\phi(\mathbf{x})$, el núcleo está dado por

$$\kappa(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}') \ , \tag{3.1}$$

siendo $\mathbf{x}$ y $\mathbf{x}'$ dos muestras en el espacio de entrada.

Una condición necesaria y suficiente para que una función $\kappa(\mathbf{x}, \mathbf{x}')$ sea un kernel válido es que la matriz de Gram sea positiva definida. La matriz de Gram es una matriz simétrica $L_{[N\times N]}$ definida como

$$L = \Phi \, \Phi^T \ , \tag{3.2}$$

donde Φ es la matriz de diseño cuya fila i-ésima viene dada por $\phi(\mathbf{x}^{(i)})^T$, i=1, 2, . . . , N.

Alto coste computacional

Obsérvese que la formulación de kernel trata con matrices cuadradas inversas N-dimensionales, mientras que el espacio inicial trata con matrices cuadradas inversas m-dimensionales, donde el número de muestras N es generalmente mucho mayor que la dimensión de las muestras m. Por lo tanto, la desventaja de la formulación de kernel es el aumento en el coste computacional, pero tiene la ventaja de trabajar directamente en términos de kernels y no con la formulación explícita del vector de características $\phi(\mathbf{x})$, lo que permite utilizar espacios característicos de alta, incluso infinita, dimensionalidad.

Transformación del kernel

Cualquier algoritmo que pueda expresarse con un producto interno en el espacio de características también puede expresarse con un kernel, lo que se denomina truco del kernel o sustitución del kernel.

Existen muchos tipos posibles de kernels, algunos de las cuales se describen a continuación con sus parámetros

$$\kappa(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}' \qquad \text{Lineal} , \tag{3.3a}$$

$$\kappa(\mathbf{x}, \mathbf{x}') = (a\mathbf{x}^T \mathbf{x}' + b)^r \qquad \text{Polinomial } (a, b, r) , \tag{3.3b}$$

$$\kappa(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) \qquad \text{Gaussiano } (\sigma) , \tag{3.3c}$$

$$\kappa(\mathbf{x}, \mathbf{x}') = \tanh(a\mathbf{x}^T \mathbf{x}' + b) \qquad \text{Sigmoidal } (a, b) . \tag{3.3d}$$

FUNCIONES DE BASE RADIAL

Las funciones de kernel que dependen de la distancia, típicamente euclídea, entre las muestras de entrada $\kappa(\mathbf{x}, \mathbf{x}') = \kappa(\|\mathbf{x} - \mathbf{x}'\|)$ se conocen como funciones de base radial, y más concretamente, funciones de base radial Gaussianas si el kernel es Gaussiano (3.3c). Téngase en cuenta que el coeficiente de normalización se omite debido a que no es una densidad de probabilidad. De aquí en adelante, las funciones de base radial serán consideradas Gaussianas.

Originalmente, las funciones de base radial se utilizaron para la interpolación exacta: dado un conjunto de muestras de entrada $\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$, el objetivo es encontrar una función suave $g(\mathbf{x})$ que sea una combinación lineal o superposición de funciones de base radial centradas en cada muestra

$$g(\mathbf{x}) = \sum_{i=1}^{N} w_i \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}^{(i)}\|^2}{2\sigma^2}\right) , \tag{3.4}$$

que se ajuste exactamente a cada valor objetivo de clase (*target*). Los pesos w_i se ajustan por mínimos cuadrados (LMS). Sin embargo, los *targets* generalmente son ruidosos y la interpolación exacta no es deseable porque produce una solución sobreajustada (*over-fitting*). Además, para grandes conjuntos de datos, es muy costoso de evaluar.

Interpolación no exacta

Una modificación consiste en reducir el número de funciones de base para proporcionar una función de interpolación suave, no exacta, en la que el número de funciones de base esté relacionado con la complejidad del problema. Ahora, los centros de las funciones de base ya no son las muestras, sino centroides representativos de las muestras; y el ancho de las funciones de base σ se convierte en diferente y entrenable para cada función de base. También es posible agregar un parámetro de sesgo para compensar la diferencia con los *targets* e incluirlo en la suma mediante una función de sesgo adicional $\phi_0 = 1$. Así, las funciones de base radiales Gaussianas resultantes son

$$\phi_k(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}_k\|^2}{2\sigma_k^2}\right) , \tag{3.5}$$

siendo $\boldsymbol{\mu}_k$ los centros de las funciones de base, k=1, 2, ..., m, con m el

número de las funciones de base Gaussianas. Téngase en cuenta que la ecuación es válida para cualquier matriz de covarianza arbitraria.

Aproximador universal

Como el perceptrón multicapa, las redes de funciones de base radial son aproximadores universales (Bishop, 1995). Incluso con restricciones leves en la forma de los núcleos, la propiedad de aproximación universal sigue siendo válida. Las redes de funciones de base radial también presentan la propiedad de ser la "mejor aproximación", consistente en la existencia de una función, de entre todas las funciones posibles con parámetros ajustables, que proporciona el mínimo error de aproximación para cualquier función dada.

Entrenamiento veloz. Dos etapas

El entrenamiento de las redes de funciones de base radial puede ser más rápido que el entrenamiento de los perceptrones multicapa, ya que es posible aplicar un procedimiento de entrenamiento de dos etapas, como se ha comentado. En el perceptrón multicapa, los parámetros entrenables generalmente se ajustan al mismo tiempo en un procedimiento de optimización global que utiliza entrenamiento supervisado. Por el contrario, en las funciones de base radial, en la primera etapa, los parámetros de las funciones de base, *i.e.*, su posición $\boldsymbol{\mu}$ y su amplitud σ en el caso de las funciones Gaussianas, se ajustan con métodos no supervisados, considerando sólo las muestras de entrada, lo que los convierte en métodos rápidos. Por lo tanto, los centros de las funciones base $\boldsymbol{\mu}$ pueden considerarse como prototipos de las muestras de entrada. En la segunda etapa del entrenamiento, los pesos de la capa de salida se ajustan resolviendo un problema lineal que minimiza una función de error, lo que también es rápido. Eso permite el uso de una gran cantidad de datos de entrenamiento no etiquetados para el cálculo de los parámetros de las funciones de base y una pequeña cantidad para el entrenamiento etiquetado de los pesos de salida, manteniendo alta la cantidad de muestras por número de parámetros para cada etapa de entrenamiento.

3.1 RBF-DTC

Se trata de un discriminante DTC no lineal formado por una red RBF Gaussiana (3.3c), que aprovecha la ventaja del pre-diseño separado de la capa oculta, con un discriminante lineal DTC en la capa de salida. Es necesaria una selección efectiva de los centroides, el parámetro que sitúa a las Gaussianas en el espacio de entrada, para proporcionar a la entrada del discriminante lineal posterior una representación apropiada con una alta capacidad expresiva de la distribución de los datos. De esta forma, la capa oculta es pre-entrenada usando técnicas de cuantificación vectorial como el Aprendizaje Competitivo Sensible a la Frecuencia (*Frequency Sensitive Competitive Learning*, FSCL), (Ahalt et al., 1990), descrito en el Apéndice C. Por último, el discriminante lineal DTC de la capa de salida es entrenado como un discriminante lineal autónomo. El discriminante no lineal construido de esta forma

preserva las propiedades del discriminante lineal del que es origen.

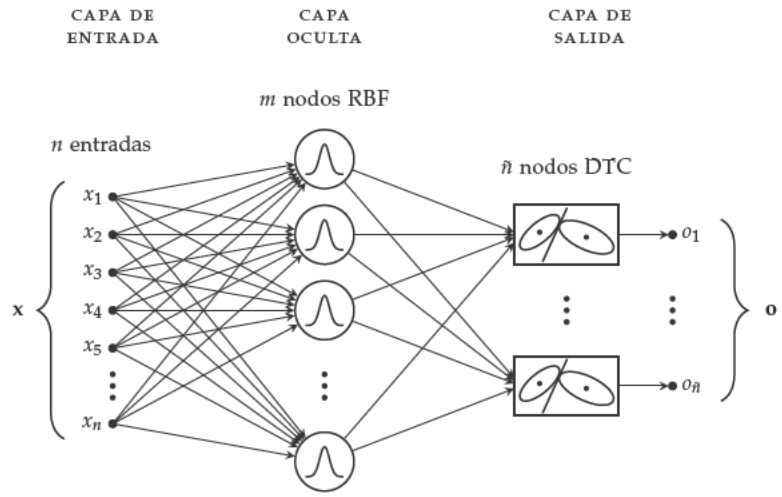


Figura 3.1: Arquitectura del discriminante RBF-DTC no lineal: patrones de entrada $\mathbf{x}$, nodos RBF en la capa oculta, nodos DTC en la capa de salida y patrones de salida $\mathbf{o}$.

La arquitectura de esta red RBF-DTC se muestra en la Figura 3.1, en la que los patrones de entrada n-dimensionales, $\mathbf{x}^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}\}$ ó $X_{[N\times n]}$ en forma matricial, i=1, 2, ..., N, siendo N el número de muestras, se transforman en N patrones ocultos m-dimensionales por medio de m funciones Gaussianas centradas en sus centroides. Cada centroide $\mathbf{c}^{(k)} = \{c_1^{(k)}, c_2^{(k)}, \ldots, c_n^{(k)}\}$, k=1, 2, ..., m, denota la posición del k-ésimo nodo Gaussiano en el espacio de entrada, con n+1 parámetros a ajustar: Los n parámetros de $\mathbf{c}^{(k)}$ y la desviación estándar. El uso de nodos Gaussianos con simetría radial proporciona la mejor opción en términos de compromiso entre el coste computacional y la precisión de clasificación (Haykin, 2009). Usando esta aproximación, la salida del nodo Gaussiano k-ésimo para la muestra de entrada i-ésima, $H_{[N\times m]}$ en forma matricial, es dada por

$$h_{i,k} = \frac{1}{\sigma_k\sqrt{2\pi}} \exp\left(-\frac{\|\mathbf{x}^{(i)} - \mathbf{c}^{(k)}\|^2}{2\sigma_k^2}\right), \tag{3.6}$$

con i=1, 2, ..., N, y k=1, 2, ..., m. En un problema multiclase, la componente j-ésima de la salida de la red para la muestra de entrada i-ésima, $O_{[N\times\tilde{n}]}$ en forma matricial, es dada por la combinación lineal de las salidas de la capa oculta y los pesos del discriminante lineal DTC, como sigue, incluyendo la regla de clasificación.

$$o_{i,j} = (\mathbf{w}^{(j)})^T \mathbf{h}^{(i)} + {\omega_0}^{(j)} \underset{C_2}{\overset{C_1}{\gtrless}} 0\,, \tag{3.7}$$

con j=1, 2, ..., $\tilde{n}$, siendo $\tilde{n}$ la dimensión de la capa de salida. Adviértase que $\tilde{n}$=1 en clasificación binaria, el caso considerado aquí.

Se pueden obtener otros discriminantes no lineales aplicando otros discriminantes lineales a la salida de la RBF, produciendo así las versiones DTC no lineales de los discriminantes DTC lineales vistos en el capítulo anterior.

Parte III

Resultados

4 Experimentos

En este capítulo se analiza el rendimiento de diferentes discriminantes lineales y no lineales en varios conjuntos de datos. Estos conjuntos de datos están compuestos por datos sintéticos con distribuciones conocidas y datos reales con distribuciones conocidas y desconocidas.

Para determinar qué algoritmo es mejor, se consideran la precisión en el conjunto de test (*Test Set Accuracy*, TSA) y el coste computacional, en términos de velocidad de entrenamiento, y los requisitos de memoria.

Test de hipótesis

Los promedios de las precisiones de test obtenidas en las repeticiones de la medición de cada algoritmo se comparan siguiendo el test de hipótesis de Newman-Keuls con un nivel de confianza del 95 %, ver Apéndice D, para muestras dependientes (Pagano, 2010), *i.e.*, se usan las mismas particiones aleatorias de los conjuntos de entrenamiento y test para evaluar cada algoritmo.

Todos los algoritmos han sido escritos y ejecutados en Matlab por los autores sin ninguna rutina ni librería externa. Existe también una versión disponible en Python.

4.1 *Conjuntos de datos*

Excepto los conjuntos de datos sintéticos, usados para el análisis de las probabilidades *a priori* en los discriminantes lineales, el resto son conjuntos de datos reales de referencia procedentes de bases de datos bien conocidas.

Las principales características de los conjuntos de datos se muestran en la Tabla 4.1, que incluye el número de muestras después de eliminar las incompletas (*missing samples*) y el número de esas muestras incompletas. Las características se escalan en el intervalo $[-1, 1]$, y los valores binarios de las clases (*target valuess*) son $\{1, -1\}$.

Bases de datos

Los conjuntos de datos usados proceden de la base de datos UCI (Lichman, 2013) (Breast cancer, Ionosphere, Heart disease, Vote, Sonar, Liver disorders, Skin segmentation), la base de datos LIBSVM (Chang y Lin, 2011) (SVM guide 1, Cod RNA), la base de datos de Two norm

	N	N_1	N_2	*Missing*	n
Two norm	7400	3703	3697	No	20
Breast cancer	683	444	239	No	10
Ionosphere	351	126	225	No	33
Heart disease	143	41	102	127	13
Vote	232	108	124	203	16
Sonar	208	111	97	No	60
Liver disorders	345	145	200	No	6
Skin segmentation	245057	50859	194198	No	3
SVM guide 1	7089	4000	3089	No	4
Cod RNA	331152	110384	220768	No	8
MNIST (0-1)	14780	6903	7877	No	400
MNIST (7-8)	14118	7293	6825	No	400
MNIST (4-5)	13137	6824	6313	No	400

Tabla 4.1: Conjuntos de datos. $N/N_1/N_2$: Total/C_1/C_2 números de las muestras después de eliminar las muestras incompletas (*missing samples*); Missing: número de las muestras incompletas eliminadas; n: número de características.

(Breiman, 1996) y la base de datos MNIST (LeCun et al., 1998). Puesto que los discriminantes comparados son clasificadores binarios, el problema MNIST se aborda a través de comparaciones de dígitos pareados. Se han seleccionado los pares (0-1), (7-8) y (5-8) porque se consideran de dificultad baja, media y alta, respectivamente.

Para la discusión de resultados, los siguientes conjuntos de datos se consideran conjuntos de datos con gran cantidad de muestras: Skin segmentation, SVM Guide 1, Cod RNA y MNIST. Los tres primeros se consideran también conjuntos de datos con una alta proporción de muestras por dimensión.

Skin segmentation, *SVM Guide 1*, *Cod RNA*, *MNIST* } $\frac{N}{n} \uparrow$ } $N \uparrow$

Cada conjunto de datos se parte en un conjunto de entrenamiento (70 %) y otro de test (30 %). Se realizan 100 ejecuciones de los algoritmos en cada conjunto de entrenamiento, aleatorizando en cada uno el orden de las muestras. Se usa la misma partición aleatoria del conjunto de datos en cada comparación entre los algoritmos, lo que se conoce como situación de muestras dependientes (Pagano, 2010).

Particiones entrenamiento y test

4.2 *Discriminantes lineales*

4.2.1 *Algoritmos a comparar*

Los discriminantes lineales usados en la comparación son:

- la formulación DTC de discriminantes lineales clásicos: MPDH-DTC, la solución directa del problema minimax MPDH, alternativa al MPM iterativo; Fisher-DTC, que añade un término independiente a la dirección proporcionada por el criterio de Fisher; y Scatter-DTC, basado en la dispersión (*scatter*) de las clases.
- un nuevo discriminante lineal DTC, llamado Quasi-Bayes-DTC, con una precisión muy cercana al óptimo de Bayes pero con menor coste computacional.
- la solución MPM lineal al problema minimax MPDH (Lanckriet et al., 2002).
- el discriminante lineal de Bayes diseñado de acuerdo al método teórico (*Theoretical Method*, TM) (Fukunaga, 1990).

4.2.2 *Comportamiento a priori*

Se analiza el rendimiento de los discriminantes lineales en un barrido de la probabilidad *a priori* P_1 de la clase C_1 en dos distribuciones sintéticas, uniforme y Gaussiana.

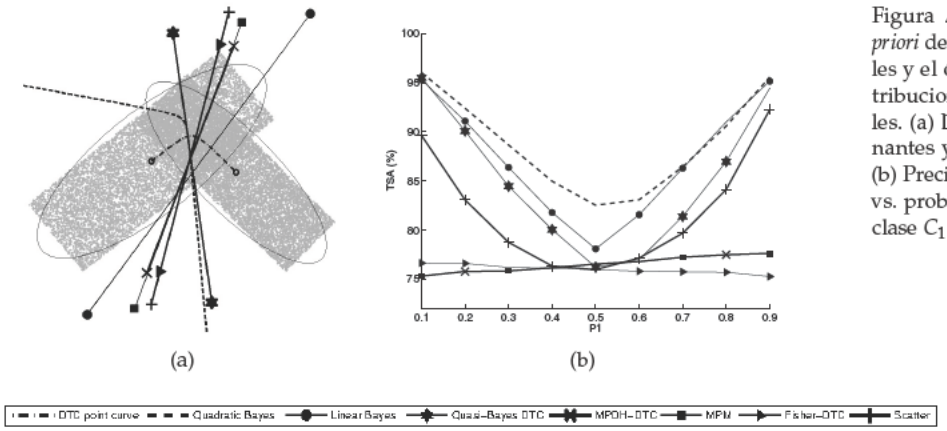


Figura 4.1: Comportamiento *a priori* de los discriminantes lineales y el óptimo de Bayes con distribuciones uniformes artificiales. (a) Distribuciones, discriminantes y curva de puntos DTC. (b) Precisión de clasificación (%) vs. probabilidad *a priori* P_1 de la clase C_1.

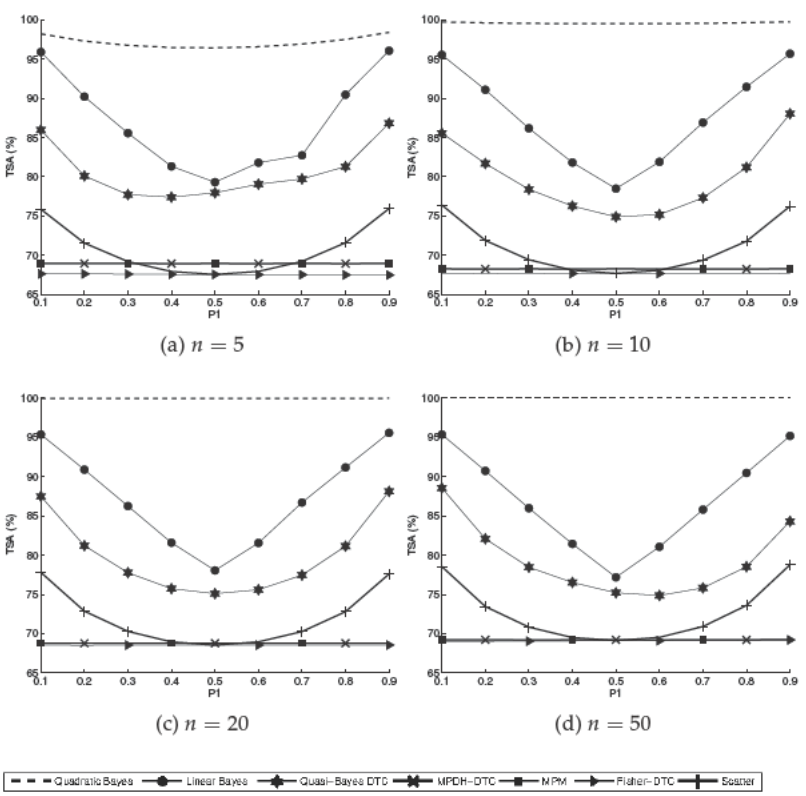


Figura 4.2: Comportamiento *a priori* de los discriminantes lineales y el óptimo de Bayes con distribuciones Gaussianas n-dimensionales artificiales. Precisión de clasficación (%) vs. probabilidad *a priori* P_1 de la clase C_1, para valores de n iguales a 5, 10, 20 y 50.

La Figura 4.1 muestra la clasificación binaria de dos distribuciones uniformes sintéticas bidimensionales. A la izquierda, las distribuciones y las elipses de sus matrices de covarianza, los discriminantes lineales, la curva de puntos DTC, *i.e.*, la curva de todos los posibles

puntos tangentes entre dos elipses centradas en las medias, y la frontera cuadrática óptima de Bayes, como referencia. Los resultados de precisión de clasificación en función de la probabilidad *a priori* P_1 de la clase C_1 se muestran a la derecha.

Los resultados de varias distribuciones gaussianas sintéticas n-dimensionales se muestran en la Figura 4.2, donde el rendimiento de los discriminantes se expresa en términos de la precisión de clasificación promediada para 100 ejecuciones.

Conjuntos de datos sintéticos

En cada ejecución, los vectores de media y las matrices de covarianza se generan aleatoriamente de manera uniforme. Las medias $\mathbf{m}_j$, $j=1,2$, están en el intervalo $[-1,1]$. Para generar las matrices de covarianza Σ_j, primero, se crea una matriz aleatoria C elemento a elemento en $[-2,2]$; segundo, la transformación para obtener una matriz de covarianza definida positiva es $\Sigma = C \cdot C^T$, donde T significa matriz transpuesta.

En ambas distribuciones uniformes y Gaussianas, los discriminantes sólo dependen de las medias y matrices de covarianza, mientras que los valores de precisión de clasificación se calculan generando 10^6 muestras de cada clase.

Bayes Lineal mejor precisión, seguido de cerca por Quasi-Bayes-DTC

En ambas distribuciones, Bayes lineal, seguido de cerca por Quasi-Bayes-DTC, proporcionan mayor precisión que el resto de los discriminantes lineales. Por otra parte, MPDH-DTC y MPM producen resultados muy similares, ambos exhiben un comportamiento minimax en el sentido de independencia de las probabilidades *a priori*. Aunque Fisher parece gráficamente similar a los discriminantes MPDH-DTC y MPM, su comportamiento no es independiente de las probabilidades *a priori*, *i.e.*, no es minimax, los resultados aparentemente planos proceden del promedio de las ejecuciones individuales de las simulaciones.

En muchos casos, la información relevante sobre el problema, como la distribuciones de las clases o las probabilidades *a priori*, es desconocida. Por tanto, la información obtenida exclusivamente de los datos puede ser errónea y los resultados pobres. Por esta razón, las soluciones minimax se emplean para minimizar el máximo riesgo posible; en otras palabras, son métodos cuya peor solución (debido a la información desconocida) es la mejor posible.

4.2.3 *Problemas de referencia*

Quasi-Bayes mejor precisión

Precisión de test (TSA). La tabla 4.2 muestra los resultados de precisión para algunos conjuntos de referencia de datos reales. Los mejores resultados globales aparecen en negrita, mientras que los mejores resultados dentro de las familias se muestran en cursiva, *i.e.*, la familia Bayesiana (comparando el discriminante lineal de Bayes con Quasi-Bayes-DTC) y la familia minimax (comparando MPM con MPDH-DTC); en ambos casos siguiendo el ya comentado test de hipótesis para las medias. De acuerdo con esto, Quasi-Bayes-DTC

produce el mejor resultado global en más casos: Skin segmentation, SVM-guide-1, Cod-RNA y MNIST (0-1), empatando con el discriminante lineal de Bayes en dos de ellos y MPDH-DTC en uno. También se puede observar que el algoritmo MPM es el mejor para Sonar. Bayes lineal, MPM y MPDH-DTC producen el mejor resultado para MNIST (7-8). En general, no es posible establecer un ganador en los casos en los que el número de muestras por dimensión es pequeño, debido a que se producen resultados con una alta dispersión y baja fiabilidad.

	Bayesiana		Minimax			
	Bayes-Lineal	Quasi-Bayes-DTC	MPM	MPDH-DTC	Fisher-DTC	Scatter-DTC
Two norm	97.8 ± 0.3	97.8 ± 0.3	97.8 ± 0.3	97.8 ± 0.3	97.8 ± 0.3	97.8 ± 0.3
Breast cancer	*97.0 ± 0.9*	95.0 ± 1.3	97.3 ± 0.9	97.2 ± 0.9	96.3 ± 1.4	97.2 ± 0.9
Ionosphere	*84.8 ± 3.3*	79.9 ± 3.9	*84.8 ± 2.9*	81.6 ± 3.3	79.0 ± 3.6	80.9 ± 3.3
Heart disease	83.4 ± 5.3	83.1 ± 5.2	81.2 ± 5.1	80.7 ± 5.1	80.9 ± 5.1	82.9 ± 5.0
Vote	96.0 ± 1.6	96.0 ± 1.6	96.0 ± 1.6	96.0 ± 1.6	96.1 ± 1.7	96.1 ± 1.8
Sonar	73.3 ± 5.8	71.9 ± 5.5	$\mathbf{77.1 \pm 5.2}$	73.6 ± 5.8	73.5 ± 5.9	73.2 ± 5.7
Liver disorders	*62.9 ± 4.6*	60.0 ± 5.3	60.9 ± 4.7	*62.9 ± 4.1*	63.1 ± 3.9	63.3 ± 3.9
Skin segmentation	93.3 ± 0.1	$\mathbf{93.6 \pm 0.1}$	93.4 ± 0.1	*93.5 ± 0.1*	91.8 ± 0.1	88.3 ± 0.1
SVM guide 1	$\mathbf{94.7 \pm 0.4}$	$\mathbf{94.7 \pm 0.4}$	93.5 ± 0.4	*94.2 ± 0.4*	86.0 ± 0.7	84.9 ± 0.7
Cod RNA	$\mathbf{95.1 \pm 0.0}$	$\mathbf{95.1 \pm 0.0}$	94.4 ± 0.0	*94.5 ± 0.1*	94.1 ± 0.1	95.0 ± 0.1
MNIST (0-1)	99.2 ± 0.2	$\mathbf{99.6 \pm 0.1}$	99.3 ± 0.1	$\mathbf{99.6 \pm 0.1}$	99.2 ± 0.2	99.3 ± 0.2
MNIST (7-8)	$\mathbf{98.2 \pm 0.2}$	98.0 ± 0.2	$\mathbf{98.1 \pm 0.2}$	$\mathbf{98.1 \pm 0.2}$	97.9 ± 0.2	98.0 ± 0.2
MNIST (5-8)	95.0 ± 0.3	94.3 ± 0.4	95.2 ± 0.3	95.1 ± 0.3	95.0 ± 0.3	95.0 ± 0.3

Tabla 4.2: Discriminantes lineales. Promedio de la precisión de test (TSA) con desviación estándar, en tanto por ciento.

Otro asunto importante es la calidad de la caracterización de las distribuciones de las clases por sus dos primeros momentos. Por ejemplo, en el problema Liver disorders, las distribuciones de las clases no están bien caracterizadas por sus dos primeros momentos, debido a que el tercer momento (*skewness*) y el cuarto (kurtosis) son elevados. Esa es la razón por la que Quasi-Bayes-DTC no obtiene un buen resultado en este problema.

MPDH-DTC mejor precisión minimax

Con respecto a los discriminantes minimax, MPDH-DTC supera a MPM en cinco casos (Liver disorders, Skin segmentation, SVM-guide-1, Cod-RNA y MNIST (0-1)) y empata en MNIST (7-8); MPM es mejor que MPDH-DTC en Ionosphere y Sonar, y no hay ninguna conclusión para los otros cuatro conjuntos de datos, donde los resultados están muy próximos. Para el caso especial del conjunto de datos MNIST, MPDH-DTC es el método que proporciona más resultados ganadores. Adviértase que el mejor rendimiento de MPDH-DTC se obtiene en los casos en los que el número de muestras por dimensión es alta; en estos casos, el problema está bien definido por los datos y un método directo como MPDH-DTC proporciona resultados más precisos que un método de optimización iterativo.

Al ser Quasi-Bayes el mejor algoritmo en rendimiento, se deduce que también es el mejor dentro de la familia Bayesiana.

	Bayes-Lineal	Todos DTCs	MPM
Two norm	20	8	**3**
Breast cancer	8	2	**1**
Ionosphere	40	9	**3**
Heart disease	10	**2**	**2**
Vote	10	3	**2**
Sonar	100	30	**10**
Liver disorders	8	2	**1**
Skin segmentation	40	**20**	**20**
SVM guide 1	6	**2**	3
Cod RNA	120	**60**	**60**
MNIST	500	240	**105**

Tabla 4.3: Discriminantes lineales. Promedio del tiempo de entrenamiento en milisegundos.

MPM mejor tiempo de entrenamiento

Tiempo de entrenamiento. La Tabla 4.3 muestra los tiempos de entrenamiento en un procesador de 2.67 GHz basado en x64 sin paralelización. La experiencia muestra que MPM necesita un número bajo de iteraciones para converger, presentando la mejor velocidad de entrenamiento. El discriminante lineal de Bayes es el algoritmo más lento en entrenamiento debido a la búsqueda del parámetro s óptimo (1.5). Por otro lado, los costes computacionales en test son similares para todos los discriminantes lineales.

4.2.4 *Discusión*

Conclusión: Quasi-Bayes ganador, MPDH-DTC mejor minimax

Como conclusión general, se puede establecer que, en el caso de que las distribuciones estén bien representadas por sus dos primeros momentos y la relación entre el número de muestras y su dimensión sea alta, Quasi-Bayes-DTC es el mejor discriminante, incluso es preferible al discriminante lineal de Bayes debido a su alto rendimiento y menor coste computacional.

En el caso de necesitar una solución minimax, MPDH-DTC es mejor que MPM en las mismas circunstancias que antes, aunque con un coste ligeramente superior.

4.3 *Discriminantes no lineales*

4.3.1 *Algoritmos a comparar*

Los discriminantes no lineales utilizados en la comparación son:

- RBFN Gaussiana con los siguientes discriminantes lineales a la salida:
  - la formulación DTC de discriminantes lineales clásicos: MPDH-DTC, Fisher-DTC y Scatter-DTC.
  - un nuevo discriminante lineal DTC, Quasi-Bayes-DTC.
  - la solución MPM lineal al problema minimax MPDH.
  - el discriminante lineal Bayesiano diseñado según el método teórico (TM).

- Gaussian-Kernel-MPM (Lanckriet et al., 2002), que proporciona una solución no lineal al problema MPDH y sirve de comparación minimax con RBF-MPM y RBF-MPDH-DTC debido a la similitud del kernel Gaussiano con los nodos RBF Gaussianos.

Primero se hace una comparación entre todos los discriminantes no lineales y luego se presenta una comparación particular en la familia Bayesiana y la familia minimax, prestando especial atención a RBF-MPDH-DTC y su alternativa, Gaussian-Kernel-MPM.

4.3.2 *Problemas de referencia*

	RBF-Bayes-Lineal	RBF-Quasi-Bayes-DTC	Gaussian-Kernel-MPM	RBF-MPM	RBF-MPDH-DTC	RBF-Fisher-DTC	RBF-Scatter-DTC
Breast cancer	60 ± 14	34 ± 12	10 ± 2	21 ± 17	27 ± 14	42 ± 20	48 ± 26
Ionosphere	52 ± 15	66 ± 9	8 ± 1	70 ± 10	70 ± 13	56 ± 17	60 ± 17
Heart disease	15 ± 9	7 ± 3	4 ± 1	18 ± 17	35 ± 13	16 ± 6	15 ± 6
Vote	54 ± 16	32 ± 10	4 ± 1	47 ± 11	52 ± 7	60 ± 8	54 ± 15
Liver disorders	46 ± 16	6 ± 7	13 ± 1	44 ± 18	46 ± 12	46 ± 16	50 ± 21
Skin segmentat.	30 ± 4	25 ± 5	11 ± 1	76 ± 4	76 ± 4	31 ± 2	30 ± 2
SVM guide 1	70 ± 8	3 ± 0	14 ± 0	76 ± 5	76 ± 6	67 ± 7	63 ± 5
Cod RNA	64 ± 20	10 ± 0	13 ± 1	13 ± 2	34 ± 14	70 ± 6	42 ± 24

Tabla 4.4: Discriminantes no lineales. Parámetros de los algoritmos. El número promedio de nodos ocultos (m) en los algoritmos RBF y la desviación estándar promedio (σ) en Gaussian-Kernel-MPM.

Parámetros. El número de núcleos RBF (explorados entre 5 y 100) y la desviación estándar (explorada entre 1 y 15) en Gaussian-Kernel-MPM se seleccionan en cada ejecución a través de validación cruzada 10-Fold. Los valores resultantes se muestran en la Tabla 4.4. La desviación estándar de cada centroide RBF se calcula heurísticamente como cuatro veces la media de las distancias euclídeas entre el centroide y sus muestras más cercanas. El entrenamiento FSCL para situar los centroides RBF incluye 100 épocas y una caída exponencial del parámetro de aprendizaje, ver Apéndice C.

Precisión de test (TSA). Para determinar qué algoritmo es mejor, los promedios de precisión de test se comparan en la Tabla 4.5. Los mejores resultados globales aparecen en negrita, mientras que los mejores resultados dentro de las familias minimax y Bayesiana se muestran en cursiva; siguiendo el test de hipótesis Newman-Keuls ya comentado.

RBF-Bayes-Lineal y RBF-MPDH-DTC mejor precisión

Excluyendo RBF-MPM, el peor algoritmo, los resultados son similares, especialmente para los primeros conjuntos de datos, los problemas con un pequeño número de muestras por dimensión. Sin embargo, en los últimos tres conjuntos de datos, en los que el número de muestras por dimensión es alto, hay una ventaja de precisión pequeña pero clara al elegir RBF-Bayes-Lineal y RBF-MPDH-DTC.

	Bayesiana		Minimax				
	RBF-Bayes-Lineal	RBF-Quasi-Bayes-DTC	Gaussian-Kernel-MPM	RBF-MPM	RBF-MPDH-DTC	RBF-Fisher-DTC	RBF-Scatter-DTC
Breast cancer	98.5 ± 0.6	98.5 ± 0.7	*98.7 ± 0.6*	97.9 ± 0.9	*98.8 ± 0.4*	99.1 ± 0.5	99.0 ± 0.5
Ionosphere	**97.1 ± 1.2**	**97.3 ± 1.3**	85.3 ± 2.2	63.9 ± 2.6	**97.5 ± 1.1**	**97.0 ± 1.2**	**97.0 ± 1.3**
Heart disease	**86.5 ± 4.3**	82.6 ± 4.7	**87.7 ± 3.6**	80.5 ± 6.4	**84.9 ± 3.8**	**85.6 ± 4.4**	**86.9 ± 3.5**
Vote	93.3 ± 1.3	93.3 ± 1.1	*93.3 ± 1.0*	85.8 ± 2.5	92.2 ± 1.6	93.4 ± 1.1	93.3 ± 1.1
Liver disorders	68.6 ± 3.6	61.3 ± 10.6	*64.5 ± 2.9*	52.7 ± 2.9	*66.5 ± 3.4*	66.1 ± 3.1	66.7 ± 4.3
Skin segmentation	**99.6 ± 0.1**	97.0 ± 0.6	**99.8 ± 0.0**	95.5 ± 0.8	**99.4 ± 0.1**	96.4 ± 8.9	97.3 ± 0.4
SVM guide 1	**96.6 ± 0.2**	89.9 ± 0.2	81.6 ± 0.8	91.0 ± 0.2	**96.4 ± 0.2**	**96.8 ± 0.1**	**96.6 ± 0.2**
Cod RNA	87.7 ± 0.3	88.0 ± 0.2	86.6 ± 0.3	80.8 ± 0.4	*87.5 ± 0.3*	87.2 ± 0.4	87.8 ± 0.4

Tabla 4.5: Discriminantes no lineales. Promedio de la precisión de test (TSA) con desviación estándar, en tanto por ciento.

Considerando los resultados minimax, la precisión de RBF-MPDH-DTC es ligeramente mejor que la de Gaussian-Kernel-MPM. Los malos resultados de RBF-MPM se deben al hecho de que la proyección de las muestras por la capa oculta aumenta la dispersión, lo que deteriora la estimación de las matrices de covarianza, de las que los parámetros internos de MPM son muy dependientes, acumulando errores en cada iteración.

Al presentar RBF-Bayes-Lineal uno de los mejores rendimientos globales, se deduce que también es el mejor dentro de la familia Bayesiana.

Tiempo de entrenamiento. Todas las simulaciones se ejecutaron en un nodo de un clúster computacional con CPU de 2.1GHz y 64GB de memoria RAM por nodo.

	RBF-Bayes-Lineal	RBF-Quasi-Bayes-DTC	Gaussian-Kernel-MPM	RBF-MPM	RBF-MPDH-DTC	RBF-Fisher-DTC	RBF-Scatter-DTC
Two norm	1331	1364	7639	7572	1262	1745	1346
Breast cancer	23	24	5	631	33	26	25
Ionosphere	25	26	3	542	23	19	18
Heart disease	5	5	2	127	5	5	5
Vote	18	15	5	460	18	18	17
Sonar	7	8	3	327	14	11	12
Liver disorders	9	10	1	326	12	12	11
Skin segment.	755	625	12819	226029	754	642	713
SVM guide 1	1629	1019	7070	6721	1259	1295	1309
Cod RNA	1825	1606	20003	318353	1772	1974	2007

Tabla 4.6: Discriminantes no lineales. Promedio del tiempo de entrenamiento en milisegundos.

Los resultados se muestran en la Tabla 4.6. Adviértase del aumento dramático del tiempo de entrenamiento en el algoritmo Gaussian-Kernel-MPM cuando el número de muestras es alto. Los malos resultados de RBF-MPM se deben a una convergencia deficiente y al consiguiente incremento de iteraciones.

Gaussian-Kernel-MPM el peor con alto número de muestras

Memoria. Mientras que RBF-MPDH-DTC tiene que lidiar con matrices cuadradas m-dimensionales, Gaussian-Kernel-MPM trata con matrices cuadradas N-dimensionales, siendo m el número de nodos ocultos y N la cantidad de muestras. Dado que el número de muestras suele ser mucho mayor que el número de nodos ocultos, un gran número de muestras dará lugar a altas necesidades memoria en Gaussian-Kernel-MPM. Se ha demostrado empíricamente que, para más de $5{\times}10^4$ muestras, Gaussian-Kernel-MPM requiere una memoria RAM muy grande, excediendo los límites de la máquina arriba descrita.

4.3.3 *Discusión*

En general, y más concretamente en problemas que están bien representados por sus dos primeros momentos y en los que la proporción de número de muestras por dimensión es alta, RBF-Bayes-Lineal y MPDH-DTC presentan los mejores resultados de rendimiento, bastante iguales entre ellos y la mejor opción dentro de sus familias. Por otro lado, RBF-MPM muestra un tiempo de entrenamiento similar con el peor rendimiento de todos, mientras que Gaussian-Kernel-MPM presenta un rendimiento similar, ligeramente peor, pero con el peor tiempo de entrenamiento.

4.4 *Coste computacional*

Los costes computacionales se muestran en la Tabla 4.7.

Lineal			No lineal			
MPM	Todos DTCs	Bayes	Gaussian-Kernel-MPM	RBF-MPM	RBF-Todos DTCs	RBF-Bayes-Lineal
$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(N^3)$	$\mathcal{O}(m^3)$	$\mathcal{O}(m^3)$	$\mathcal{O}(m^3)$

Tabla 4.7: Coste computacional de los algoritmos lineales y no lineales. N: número de muestras de entrenamiento; n: número de características; m: número de nodos RBF.

Discriminantes lineales. En los discriminantes lineales, aunque el coste computacional de DTC parece ser ligeramente mayor que el coste de MPM, es necesario considerar que el valor de MPM no incluye el coste requerido para determinar los valores óptimos de algunos parámetros internos del algoritmo, como el parámetro de regularización de la matriz hessiana y el parámetro de tolerancia de detención del entrenamiento, que establece que la suma de dos variables, β_k y η_k, debe ser lo suficientemente pequeña (Lanckriet et al., 2002). Por lo tanto, es necesario ajustarlos por validación cruzada. El coste de DTC se calcula considerando el coste de resolver las raíces del polinomio de Clark de grado $2n$, $\mathcal{O}((2n)^3)$, y el coste de invertir matrices cuadradas n-dimensionales, $\mathcal{O}(n^3)$; mientras que MPM necesita resolver un sistema lineal de matriz cuadrada de orden n por mínimos cuadrados (LMS), con un coste de $\mathcal{O}(n^3)$ por cada

Lineales: coste similar

iteración. El coste del algoritmo iterativo para buscar el mínimo error de Bayesiano en el discriminante lineal de Bayes incluye el cálculo de inversas de matrices cuadradas de orden n, tantas como el número de pasos Δs en los que se divide el intervalo $[0, 1]$ de búsqueda de s.

Discriminantes no lineales. En los algoritmos lineales DTC, el principal coste es resolver las raíces del polinomio de Clark de orden el número de características de las muestras de entrada al DTC, por lo que, en el caso no lineal, ese coste depende de m, el número de nodos RBF.

En Bayes, el coste es similar a DTC. Esto se debe a que la carga computacional de resolver el polinomio de Clark de DTC es prácticamente la misma que la carga requerida por la búsqueda iterativa del error mínimo en el método de Bayes.

En los algoritmos RBF (*i.e.*, todos excepto Gaussian-Kernel-MPM), también existe el coste asociado al entrenamiento FSCL de los centroides, que no está incluido porque es cuadrático respecto al número de características, y por lo tanto es menor que el coste polinómico de DTC de orden cúbico con respecto al número de nodos RBF.

Gaussian-Kernel-MPM el peor con alto número de muestras

Por otro lado, Gaussian-Kernel-MPM resuelve un sistema lineal de matriz cuadrada de orden N por mínimos cuadrados (LMS) con un coste de $\mathcal{O}(N^3)$. Esa es la razón por la que, en el caso no lineal, si el número de muestras es muy alto, el coste computacional de Gaussian-Kernel-MPM aumenta dramáticamente en comparación con DTC. Además, Gaussian-Kernel-MPM presenta un coste adicional, al igual que MPM, debido a los parámetros internos que deben fijarse por validación cruzada antes del entrenamiento, es decir, el parámetro de regularización del Hessiano y el parámetro de tolerancia de detención del entrenamiento.

Teóricamente, la red RBF-MPM presenta un coste similar a RBF-MPDH-DTC ya que sus discriminantes lineales de salida tienen un coste similar, como se muestra anteriormente. Sin embargo, existe una diferencia en los resultados de tiempo de entrenamiento debida a la mala convergencia ya explicada anteriormente.

4.5 Conclusiones

En los algoritmos lineales, es preferible Quasi-Bayes-DTC, especialmente en problemas con un gran número de muestras por dimensión, incluso más que el discriminante lineal de Bayes, debido a su alto rendimiento y menor coste computacional, ya que es una solución no iterativa; a menos que se requiera una solución minimax, en ese caso se prefiere MPDH-DTC.

Con respecto a los algoritmos no lineales, la solución Bayesiana de RBF-Bayes-Lineal y la solución minimax de RBF-MPDH-DTC tienen globalmente un rendimiento bastante parecido, con un tiempo

	Lineal				No lineal				
	Bayesiana		Minimax		Bayesiana		Minimax		
	Bayes-Lineal	Quasi-Bayes-DTC	MPM	MPDH-DTC	RBF-Bayes-Lineal	RBF-Quasi-Bayes-DTC	Gaussian-Kernel-MPM	RBF-MPM	**RBF-MPDH-DTC**
Precisión	✓	✓✓	✗✗✗	✗✗	✓✓✓	✗✗	✓✓	✗✗✗	✓✓✓
Coste computacional	✓	✓✓	✓✓✓	✓✓	✓	✓	✗✗✗	✓	✓

✓: buen desempeño. ✗: mal desempeño.

Tabla 4.8: Conclusiones para los principales algoritmos.

de entrenamiento ligeramente mejor para RBF-MPDH-DTC, lo que lo hace preferible porque es minimax también. El mejor algoritmo minimax es RBF-MPDH-DTC, con mejor tiempo de entrenamiento que Gaussian-Kernel-MPM y mejor precisión que RBF-MPM.

En general, el tiempo de entrenamiento de los algoritmos lineales es muy bajo pero con la desventaja de una menor precisión, excepto en problemas que son linealmente separables en su origen.

Considerando todos los aspectos, RBF-MPDH-DTC es la mejor opción debido a su alta precisión con un coste computacional competitivo. Además, con la ventaja de proporcionar una solución minimax, que puede ser útil en el caso de que las frecuencias de las clases en el entrenamiento no sean representativas de las probabilidades *a priori* reales.

Parte IV

Apéndices

A

Acotación del error de clasificación

Se trata de un problema clásico de estadística consistente en acotar la probabilidad de que una variable aleatoria pertenezca a un determinado conjunto (el de clasificación errónea en este caso) dada la información de algunos de sus momentos. Cabe destacar que la solución de este problema es la acotación de la probabilidad, independientemente de que exista una distribución en la que se alcance esa cota.

El problema se denomina acotación-(n, k, Ω), donde n es la dimensión de los datos, k el número de momentos conocidos y Ω el conjunto donde se acota la probabilidad de que la variable aleatoria pertenezca a él.

Existen distintas soluciones que ofrecen aproximaciones basadas en optimización semidefinida (Bertsimas y Popescu, 2005) para calcular las cotas según los valores de n, k y Ω

Sólo consideramos los dos primeros momentos $(n, 2, \mathbb{R}^n)$ para problemas multidimensionales. Se pueden considerar más momentos, $k > 2$, y sus formulaciones correspondientes para acotar la probabilidad de error, pero conforme aumenta la dimensión de los datos y el número de momentos momentos, el cálculo es más difícil. Tal vez los dos primeros momentos no consigan una acotación óptima pero es suficiente y es fácil extraer conclusiones gráficas gracias a los elipsoides de probabilidad.

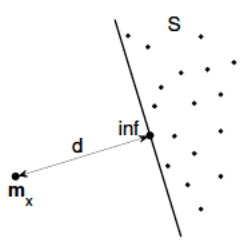


Figura A.1: Marshall. Acotación de la probabilidad de que un vector aleatorio x pertenezca al conjunto S dados los dos primeros momentos. Distancia de Mahalanobis d desde el ínfimo de distancias hasta la media $\mathbf{m}_x$.

En primer lugar definimos la calidad de las cotas encontradas. Se define la mejor posible o cota superior estricta γ de $P(\mathbf{x} \in S)$ como

$$\gamma = \sup_{\mathbf{x} \sim \Pi} P(\mathbf{x} \in S) \,, \tag{A.1}$$

donde $\Pi = (M_1, M_2, \ldots, M_k)^T$ es una secuencia de k momentos factibles, $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ definida en $\Omega \subseteq \mathbb{R}^n$ tiene una distribución factible en Π, escrito como $\mathbf{x} \sim \Pi$, y, por último, $S \subseteq \Omega$, es un conjunto semialgebraico, es decir, definido en términos de desigualdades de polinomios.

Adviértase que una cota puede ser estricta sin tener que ser alcanzable, sólo es necesario que lo sea asintóticamente.

En segundo lugar, para calcular las cotas se hace uso, en el caso de una variable, de las desigualdades de Markov $(1, 1, \Omega)$, Chebyshev $(1, 2, \Omega)$ y Chernoff, si se conocen el primer momento, los dos primeros momentos o todos los momentos (la función generadora) de una variable aleatoria, respectivamente. Estas soluciones son factibles pero no necesariamente soluciones óptimas o cotas estrictas del problema de acotación (n, k, Ω).

Marshall y Olkin (1960) muestran que es posible calcular explícitamente las cotas de las probabilidades, dados los dos primeros momentos $(n, 2, \mathbb{R}^n)$, generalizando la desigualdad de Chebyshev para un escenario multivariable, la situación que nos interesa

$$\sup_{\mathbf{x}\sim(\mathbf{m}_x,\Sigma_x)} P\{\mathbf{x} \in S\} = \frac{1}{1+d^2} \,, \tag{A.2}$$

con

$$d^2 = \inf_{\mathbf{x}\in S} (\mathbf{x}-\mathbf{m}_x)^T \Sigma_x^{-1} (\mathbf{x}-\mathbf{m}_x) \,, \tag{A.3}$$

donde $\mathbf{x}$ es un vector aleatorio, S es un espacio-mitad convexo, *i.e.*, separa el espacio en dos conjuntos convexos, uno de ellos es él mismo. La distancia de Mahalanobis d es una medición multidimensional de cuántas desviaciones estándar dista un punto de la media. Puesto que se tiene en cuenta la matriz de covarianza de la distribución, las distancias de Mahalanobis corresponden a superficies de nivel elipsoidales, en lugar de las esféricas de las distancias euclídeas, en las que la matriz de covarianza es la identidad. La frontera de S, debido a que es un conjunto convexo, es tangente a una determinada superficie de nivel y el punto tangente es el ínfimo de todas las distancias de Mahalanobis desde la media a todos los puntos de S. Adviértase que el ínfimo es siempre único si existe. La probabilidad de formar parte de S decrece con la distancia d porque a mayor distancia de la media a la frontera de S, más pequeña es la región S y menos puntos caen en ella. Por lo tanto, dado una frontera fija, el supremo de la probabilidad se alcanza con el ínfimo de las distancias. La solución obtenida es válida para todas las distribuciones posibles de $\mathbf{x}$ dados los mismos dos primeros momentos, sin tener que asumir Gaussianidad. Se muestra un ejemplo de juguete en la Fig. A.1.

En clasificación binaria, S corresponde a la región de clasificación errónea. La cota superior de la probabilidad de clasificación incorrecta o probabilidad de error para nuevas muestras, *i.e.*, que caigan en la región S, es descrita por (A.2). El objetivo será minimizar esa probabilidad de clasificación incorrecta.

	Precisión		
Distribuciones	Total	C_1	C_2
Gaussiana	94.79	94.8	94.8
t	95.57	95.5	95.7
Uniforme	98.68	98.6	98.7

Tabla A.1: Ejemplo de juguete: precisión de acierto (%) en distribuciones, con igual vector de media y matriz de covarianza por clase, dada la cota de acierto $\varepsilon = 75\,\%$. Número de muestras $N = 10^5$

B

Criterio Minimax

Minimax significa maximizar primero y minimizar después o viceversa, minimizar primero y maximizar después (principio de dualidad).

El objetivo de minimax es minimizar el peor caso o máximo riesgo. En clasificación consiste en buscar el discriminante que minimice el máximo error de cada posible clasificador. De esta forma, hay dos riesgos a minimizar: el primero, el error de clasificación producido por el cambio en las probabilidades *a priori* respecto al entrenamiento y, el segundo, el desconocimiento de la distribución de los datos y las posibles asunciones acerca de ellas y, como caso especial, de sus probabilidades *a priori*.

La Figura B.1 muestra en la curva cóncava de trazo continuo la probabilidad de clasificación errónea de el discriminante óptimo de Bayes para cada valor de la probabilidad *a priori* de la clase C_1. Los puntos A y B son dos ejemplos de la probabilidad de error óptima de Bayes para dos probabilidades *a priori* dadas. Una vez que el clasificador de Bayes es entrenado para una probabilidad *a priori* dada, *e.g.*, el punto B, si esa probabilidad *a priori* cambia, el error de clasificación es lineal con el desplazamiento de la probabilidad *a priori* y tangente con la curva óptima de Bayes en el punto de partida, como se muestra en la línea de trazo discontinuo y los puntos B^- y B^+ para dos desplazamientos dados. La solución minimax corresponde al punto A, que es el caso peor de la probabilidad de clasificación errónea de Bayes, *i.e.*, el mayor error, pero también es el caso peor mínimo, para cualquier desplazamiento de la probabilidad *a priori*, *e.g.*, menor que B^+. Por lo tanto, la solución minimax minimiza el caso peor. Una consecuencia de la solución minimax es la independencia de las probabilidades *a priori*, como muestra la línea horizontal.

Una consecuencia de la independencia de las probabilidades *a priori* en la solución minimax es que la probabilidad de error (o de acierto) de cada clase es igual. El error total es una ponderación del error de cada clase multiplicado por las probabilidades *a priori*. Por lo tanto, es equivalente que el error de cada clase sea el mismo y la independencia de las probabilidades *a priori* en el error total.

Ejemplo sencillo:

Resolver maximizando la suma, sujeta a una restricción, siendo cada incógnita la menor posible.

$$\max_{min\ x,y,z} \quad x+y+z$$

$$s.t. \quad x+y+z \leq 16$$

La solución es una combinación de 5, 5, y 6. Si no fuese minimax, la solución podría ser 14, 1 y 1.

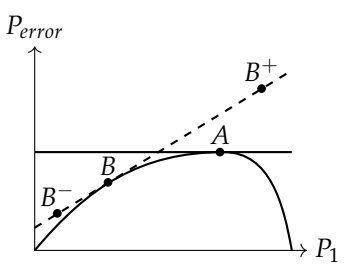
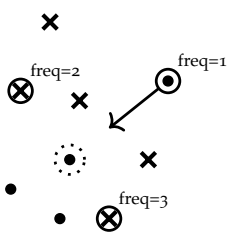


Figura B.1: Comportamiento minimax con probabilidades *a priori*. Horizontal: probabilidad *a priori* de la clase C_1. Vertical: Probabilidad de error de clasificación.

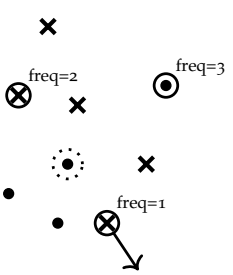
C

Aprendizaje Competitivo Sensible a la Frecuencia (FSCL)

Los centroides se entrenan con el algoritmo de Aprendizaje Competitivo Sensible a la Frecuencia (*Frequency Sensitive Competitive Learning*, FSCL) de Ahalt et al. (1990). Este es un procedimiento de aprendizaje competitivo que propone una nueva medida de distancia al multiplicar el número de veces que una neurona gana a la función de distancia original. De esta manera, una neurona –en nuestro caso, los centroides de la RBF– que frecuentemente gana en la competición para obtener una muestra, tendrá una menor probabilidad de ganar la próxima vez. Está comprobado que este mecanismo converge a las posiciones de las neuronas que maximizan la entropía, lo cual es de gran importancia porque los centroides seleccionados representarán adecuadamente la población de muestras de entrenamiento, evitando los riesgos de representación insuficiente o excesiva de partes de la población. También permite estimar las desviaciones estándar Gaussianas que promedian la distancia desde cada centroide a sus muestras más cercanas. El número de centroides, que también es la dimensión de la capa oculta, se selecciona mediante validación cruzada (CV), por lo que es diferente para cada algoritmo y conjunto de datos.



(a) Igual clase



(b) Distinta clase

Figura C.1: Algoritmo FSCL. Movimiento de centroides. Para cada muestra, el centroide más cercano (distancia multiplicada por la frecuencia de éxito) se mueve hacia la muestra si son de la misma clase y en dirección opuesta si son de clases distintas. Las muestras son las cruces y los puntos, los centroides son los círculos y la muestra analizada está dentro del círculo punteado.

Aprendizaje competitivo FSCL:

- Para cada muestra:
    1. Búsqueda del centroide más cercano a la muestra
        - distancia multiplicada por la frecuencia ganadora
    2. Movimiento del centroide ganador
    3. Incremento de la frecuencia ganadora

C.1 Algoritmo

Algoritmo 3: Aprendizaje Competitivo Sensible a la Frecuencia (*Frequency Sensitive Competitive Learning*, FSCL)

Datos: $X_{[N\times n]}$ patrones etiquetados, N_c: núm. de centroides
Resultado: $C_{[N_c\times n]}$ centroides

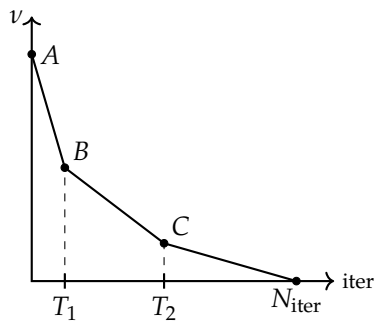
$N_e = 100$ `% Número de épocas`
$N_{\text{iter}} = N \cdot N_e$ `% Número de iteraciones`

`% Cálculo del parámetro de aprendizaje` ν
$A = 0.3$
if *caída lineal* **then**
  $B = 0.15;\ C = 0.05$
  $T_1 = 0.125 \cdot N_{\text{iter}};\ T_2 = 0.5 \cdot N_{\text{iter}};\ T_3 = N_{\text{iter}}$
  $\nu[1:T_1] = -\frac{A-B}{T_1}[1:T_1] + A$
  $\nu[T_1:T_2] = -\frac{B-C}{T_2-T_1}\left([T_1:T_2] - T_1\right) + B$
  $\nu[T_2:T_3] = -\frac{C}{T_3-T_2}\left([T_2:T_3] - T_2\right) + C$
else if *caída exponencial* **then**
  $\tau = 2/N_{\text{iter}}$ `% Constante de tiempo`
  $\nu = \frac{A}{e^{-\tau} - e^{-\tau\cdot N_{\text{iter}}}}\left(e^{-[1:N_{\text{iter}}]} - e^{-\tau\cdot N_{\text{iter}}}\right)$

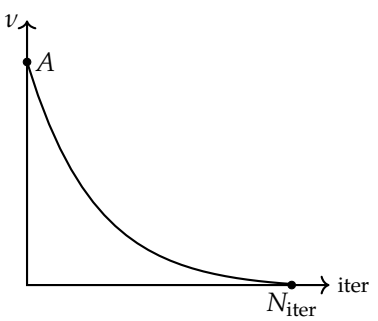
`% Cálculo de los centroides` ν
$\text{freq}_m \leftarrow 0,\ m = 1,2,\ldots,N_c$ `% Inicialización frecuencias`
$C_m \leftarrow \text{rand}$ `% Inicialización aleatoria posiciones`
for $e = 1$ **to** N_e **do**
  foreach X_i, i=1,2,...,N **do**
    `% j: índice del centroide más cercano a` X_i
    $j \leftarrow \arg\min_m\{\text{freq}_m \cdot \|X_i - C_m\|\}$

    `% Desplazamiento del centroide ganador` C_j
    $k = e + i$
    if X_i *igual clase que* C_j **then**
      $C_j \leftarrow C_j + \nu_k\left(X_i - C_j\right)$
    else
      $C_j \leftarrow C_j - \nu_k\left(X_i - C_j\right)$

    `% Actualización de la frecuencia ganadora`
    $\text{freq}_j \leftarrow \text{freq}_j + 1$



(a) Caída lineal por tramos



(b) Caída exponencial

Figura C.2: Algoritmo FSCL. Caída del parámetro de aprendizaje ν

D

Test de Hipótesis

El objetivo de un experimento científico es la validación de una hipótesis. La hipótesis más común para probar es si el rendimiento de un método es mejor que otro.

El primer paso de la metodología consiste en seleccionar una muestra representativa de la población a la que se generalizará la hipótesis. La forma de validar la hipótesis es dividirla en dos: la hipótesis alternativa (H_1) y la hipótesis nula (H_0). La hipótesis alternativa explica que la diferencia de resultados en los diferentes métodos se debe a un factor real, *i.e.*, la causa es la variable independiente. Por el contrario, la afirmación nula de la hipótesis de que esa diferencia se debe únicamente a factores de azar (Pagano, 2010).

El procedimiento de test de hipótesis en la inferencia estadística es el siguiente: Primero, supongamos que la hipótesis nula es verdadera y evaluamos que la probabilidad del azar sea la causa de la diferencia de resultados; segundo, comparar la probabilidad obtenida con un nivel de umbral llamado probabilidad crítica (α). Si es mayor que la probabilidad crítica, se retiene la hipótesis nula concluyendo que la diferencia de los resultados se debe únicamente al azar, *i.e.*, ambos métodos producen los mismos resultados. Por el contrario, si es menor, se rechaza la hipótesis nula y se acepta la hipótesis alternativa, *i.e.*, la diferencia de los resultados se debe a la variable independiente, que es diferente en ambos métodos, y entonces uno es mejor que el otro.

Existen dos tipos de errores derivados de una decisión incorrecta, ver la Tabla D.1: El error de Tipo I, como resultado de la decisión de rechazar la hipótesis nula cuando la hipótesis nula es cierta; y el error Tipo II cuando se retiene una hipótesis nula falsa.

El nivel de probabilidad crítica (α) es el límite del error de Tipo I, por lo que disminuir α disminuye el error de Tipo I, pero aumenta el error de Tipo II. En la fabricación y presentación de nuevos descubrimientos científicos es mejor disminuir el error de Tipo I seleccionando un valor bajo de α, generalmente 0.01 o 0.05. Sin embargo, en el caso de la exploración de nuevas posibilidades, es más importante reducir

Muestra o grupo. Conjunto de individuos evaluados en una determinada **condición**, *e.g.*, un algoritmo, método o un placebo

Individuo o sujeto. *e.g.*, una partición aleatoria de un conjunto de datos

Variable dependiente. El efecto del experimento, las mediciones

Variable independiente. La causa (no aleatoria) de la variable dependiente

Hipótesis alternativa. Afirma que las diferencias en los resultados se deben a la variable independiente

Hipótesis nula. Opuesto lógico a la hipótesis alternativa. Afirma que la variable independiente no tiene efecto en la variable dependiente, el azar es la única causa

Nivel crítico de probabilidad. La hipótesis nula se rechaza si la probabilidad de que sólo el azar sea la causa de los resultados es igual o menor que este nivel

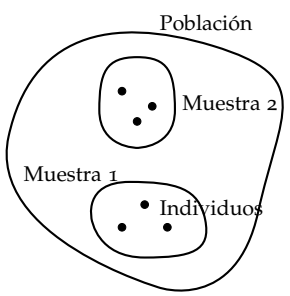


Figura D.1: Ejemplo de una población y 2 muestras con $N = 3$ individuos.

el error de Tipo II, por lo que se consideran valores más altos de α, como 0.10 o incluso 0.20.

La potencia (*power*) mide la sensibilidad del experimento de detectar un efecto real, el efecto que la variable independiente produce en la variable dependiente. También es la probabilidad de rechazar la hipótesis nula correctamente, por lo tanto, es deseable que tenga un valor alto, lo más cercano posible a 1; sin embargo, es difícil ver valores más altos que 0.8, es común de 0.4 a 0.60. La potencia depende del tamaño del efecto real, cuanto mayor sea la contribución de la variable independiente, mayor será la potencia. Aumentar el tamaño de la muestra (N) también aumenta la potencia.

$$\beta = 1 - \text{potencia} \; . \tag{D.1}$$

	Realidad	
Decisión	H_0 verdadera	H_0 falsa
Retener H_0	0	Tipo II (β)
Rechazar H_0	Tipo I (α)	0

Tabla D.1: Errores al aceptar y rechazar H_0 incorrectamente

Dependencia de los errores

$\alpha \downarrow$ { Error Tipo I $\downarrow$ / Error Tipo II (β) $\uparrow$

$N \uparrow$ / tamaño del efecto real $\uparrow$ } Potencia $\uparrow$, $\beta \downarrow$

Completando el método de test de hipótesis, el primer paso es el cálculo del estadístico (*statistic*), *e.g.*, la media de los rendimientos de precisión utilizados para comparar algoritmos o métodos; segundo, suponiendo que la hipótesis nula es nula, calcular la probabilidad de azar de los resultados obtenidos con la distribución muestral (*sampling distribution*) del estadístico; y, finalmente, comparar esa probabilidad de azar solo con la probabilidad crítica de rechazar o no la hipótesis nula.

Test de hipótesis

1. Cálculo del estadístico (*e.g.*, media)
2. Cálculo de la distribución muestral (probabilidad de azar)
3. P. azar { $> \alpha \Rightarrow$retener H_0 / $\leq \alpha \Rightarrow$rechazar H_0 (aceptar H_1)

D.1 Test estadísticos

La distribución muestral de un estadístico proporciona las probabilidades de los valores que el estadístico puede tomar si son causadas sólo por azar, es el paso previo a la comparación con la probabilidad crítica α, como se mostró anteriormente.

La población de hipótesis nula (*null-hypothesis population*) es un conjunto real o teórico que resulta de realizar el experimento en toda la población en el caso de que la variable independiente no tenga ningún efecto. Se utiliza para probar la hipótesis nula H_0 tomando muestras de N individuos, tantas como combinaciones de N individuos existan. El estadístico, *e.g.*, la media, se evalúa en cada muestra, y el conjunto de todos estos estadísticos forma la población de hipótesis nula. Dependiendo del estadístico elegido y de la caracterización de la distribución muestral, existen diferentes test estadísticos, los más importantes se muestran a continuación.

D.2 Test-z de desviación normal

En este test, el estadístico es la media y la distribución muestral del estadístico es una distribución normal.

Los requisitos para usar este test son: Los parámetros de la población de hipótesis nula (μ, σ) son conocidos; la distribución muestral de la media es normal, lo que ocurre cuando la distribución de hipótesis nula es normal o cuando la distribución de hipótesis nula es

Requisitos del Test-z

- Pob. H_0 (μ, σ) conocidos
- Dist. muest. medias ($\mu_{\bar{X}}$, $\sigma_{\bar{X}}$) es normal { Pob. H_0 es normal *ó* / $N \geq 30$
- Una muestra

casi normal y el número de observaciones de la muestra es $N \geq 30$, según el Teorema del Límite Central; y sólo hay una muestra, por lo que la hipótesis nula afirma que la muestra es una muestra aleatoria de la población de hipótesis nula y la hipótesis alternativa afirma lo contrario.

Los parámetros de la distribución muestral son la media ($\mu_{\bar{X}}$) y la desviación estándar ($\sigma_{\bar{X}}$), con la siguiente relación con los parámetros de la población de hipótesis nula:

$$\mu_{\bar{X}} = \mu \,, \tag{D.2a}$$

$$\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{N}} \,, \tag{D.2b}$$

D.3 Test-t de Student

Sólo la media se ve afectada por la variable independiente, no la varianza.

D.3.1 Una muestra

Requisitos test-t de una muestra

- Pob. H_0 (μ, ~~σ~~) conocidos
- Dist. muest. medias ($\mu_{\bar{X}}$, $\sigma_{\bar{X}}$) es normal $\begin{cases} \text{Pob. } H_0 \text{ es normal } \acute{o} \\ N \geq 30 \end{cases}$
- Una muestra

Es similar al test-z, pero aquí se presenta un caso más común, una población de hipótesis nula de la cual se conoce la media pero no la desviación estándar, esta es la razón por la que la distribución normal no se usa como la distribución muestral; en cambio, se usa la distribución t como la distribución muestral del estadístico, la media. Dado que σ es desconocido, se calcula a partir de la muestra y es llamada s. Así, los parámetros de la distribución muestral son los siguientes:

$$\mu_{\bar{X}} = \mu \,, \tag{D.3a}$$

$$s_{\bar{X}} = \frac{s}{\sqrt{N}} \,, \tag{D.3b}$$

siendo $s_{\bar{X}}$ una estimación $\sigma_{\bar{X}}$.

La distribución t se parece a la distribución normal, con la diferencia de que son una familia de curvas que depende del tamaño N de la muestra. Más precisamente, la distribución t varía con los grados de libertad (df) de la muestra, *i.e.*, el número de individuos que pueden variar libremente al calcular el estadístico.

$$df = N - 1 \,, \tag{D.4}$$

si $df = \infty$ la distribución t es igual a la distribución normal.

D.3.2 Dos muestras

Este es el caso llamado experimento de dos condiciones, dos grupos o dos muestras, entendiendo que el mismo conjunto de individuos evaluados en dos condiciones diferentes constituye dos muestras diferentes, ya que las poblaciones de las que provienen son diferentes.

Grupos correlados. En los grupos correlados, cada sujeto se evalúa en dos condiciones diferentes. También es posible utilizar diferentes sujetos emparejados en pares que compartan las mismas características, *e.g.*, edad o género, etc. La hipótesis nula afirma que la diferencia de resultados en las condiciones es una muestra aleatoria de una población de diferencias con media cero, *i.e.*, no hay diferencia entre los resultados. Por lo tanto, los parámetros de la distribución muestral del estadístico, que es la media de las diferencias de los resultados, son $\mu_D = 0$ y σ_D (desconocido, pero no es necesario para el test-t).

Req. test-t dos grupos corr.

- Param. pob. H_0 ($\mu_D = 0$, ~~σ_D~~)
- Dist. muestral de la media de las diferencias (μ_D, σ_D) es normal $\begin{cases} \text{Pob. } H_0 \text{ es normal } \acute{o} \\ N \geq 30 \end{cases}$

Grupos independientes. Los sujetos se seleccionan al azar de la población de sujetos. Hay dos poblaciones de hipótesis nula. La estadística es la diferencia de las medias de cada grupo.

La homogeneidad de la varianza consiste en asumir que las varianzas de las dos poblaciones son iguales $\sigma_1^2 = \sigma_2^2$. Sin embargo, la robustez del test-t hace que sea relativamente insensible a las violaciones de la normalidad y la homogeneidad de la varianza. Pero si las varianzas son muy diferentes, puede que la variable independiente no tenga el mismo efecto en ambas poblaciones.

Se pierde un grado de libertad cada vez que se estima una desviación estándar

$$df = N - 2\,, \tag{D.5}$$

donde $N = n_1 + n_2$, siendo n_1 y n_2 los tamaños de cada muestra.

Req. t-test dos grupos indep.

- Param. dos pob. H_0 ($\mu_1 - \mu 2 = 0$, ~~σ_1~~ $-$ ~~σ_2~~ $= 0$)
- Dist. muestral de la diferencia de medias $\bar{X}_1 - \bar{X}_2$ es normal $\begin{cases} \text{Pob. } H_0 \text{ son normales } \acute{o} \\ n_1 \geq 30 \;\&\; n_2 \geq 30 \end{cases}$
- Homogeneidad var. $\sigma_1^2 = \sigma_2^2$

Correlado vs. independiente Aunque en cada problema una elección u otra puede ser más natural, es necesario considerar algunos aspectos. En los grupos correlados, la variabilidad de la diferencia de resultados es menor, lo que aumenta la potencia. Mientras que los grados de libertad en los grupos independientes son mayores, lo que disminuye el valor crítico de t.

Grupos correlados
variabilidad de la muestra $\downarrow \Rightarrow$ potencia $\uparrow$

Grupos independientes
$df \uparrow \Rightarrow t_{\text{crit}} \downarrow \Rightarrow$ Error Tipo I $\uparrow$

D.4 Múltiples muestras

A diferencia del test anterior, la media no se usa como estadístico sino la varianza, con la distribución F como la distribución muestral, que es una relación entre dos estimaciones independientes de la varianza poblacional. Es apropiado para el análisis de más de dos muestras y las condiciones pueden ser el efecto de diferentes variables independientes o el rango de la variable independiente. La razón para no hacer comparaciones múltiples con el test anterior para dos muestras entre pares de condiciones es que múltiples evaluaciones t o z aumentan el error de Tipo I. A pesar de que el test-F analiza la varianza, permite hacer una comparación sobre todas entre las medias de los grupos evitando el aumento de la probabilidad de error de Tipo I. Se

El test-F evita el aumento del error de Tipo I en comparaciones múltiples pero no especifica qué condición es mejor

usa en grupos de medidas independientes y repetidas y cuando se investigan dos o más factores.

Al igual que el test-t, se supone que sólo la media se ve afectada por la variable independiente, no la varianza.

La hipótesis nula afirma que todas las condiciones tienen el mismo efecto en la variable dependiente, mientras que la hipótesis alternativa, que siempre es no direccional, afirma que una o más condiciones tienen efectos diferentes.

El test-F es la relación entre dos estimaciones de varianza de la población de hipótesis nula: la varianza entre grupos y la varianza dentro de los grupos. Cuanto más alto sea el efecto de la variable independiente, mayor será la varianza entre grupos, mientras que la varianza dentro de los grupos permanece igual porque cada grupo recibe el mismo nivel de variable independiente. Por lo tanto, cuanto mayor sea el efecto de la variable independiente, mayor sera el valor F y la probabilidad de rechazar la hipótesis nula.

$$F_{\text{obt}} = \frac{s_B^2}{s_W^2} \, , \tag{D.6}$$

siendo s_B^2 la variación entre grupos y s_W^2 la variación dentro de grupos. Adviértase que si $F_{\text{obt}} \leq 1$ la explicación es el azar y la hipótesis nula se mantiene. En el caso de dos grupos independientes, el test-t y el test-F están relacionados por $t^2 = F$.

Los supuestos del análisis de varianza para k grupos son similares a los del test-t para grupos independientes. Al igual que el test-t, el test-F es robusto. Se ve afectado poco por las poblaciones que son casi normales y es relativamente insensible a las violaciones de la homogeneidad de la varianza. Al igual que en los test anteriores, la potencia aumenta con N, es mayor para efectos grandes de la variable independiente y cuanto menor es la variabilidad de la muestra, mayor es la potencia para detectar el efecto real.

Requisitos del test-F

- Las poblaciones de las muestras son normales
- Homogeneidad varianza $\sigma_1^2 = \sigma_2^2 = \ldots = \sigma_k^2$

Cuando se hacen comparaciones múltiples, es necesario corregir el incremento de la probabilidad del error de Tipo I. Hay dos métodos principales que mantienen la tasa de error de Tipo I en α al hacer todas las comparaciones posibles: el HSD (Honestly Significant Difference) de Tuckey y el test de Newman-Keuls. Estos métodos utilizan las distribuciones Q o *Studentializada*.

Potencia

$N \uparrow$
tamaño del efecto real $\uparrow$
variabilidad muestra $s_W^2 \uparrow$
} Potencia $\uparrow$

D.5 Test de Newman-Keuls

La diferencia con el test-F es que su hipótesis alternativa afirma que una o más condiciones son diferentes entre sí, pero no especifica cuáles porque no realiza comparaciones múltiples. En cambio, las pruebas de Newman-Keuls hacen comparaciones múltiples que establecen al ganador o ganadores en una lista de medias ordenada por rango.

Newman-Keuls evita el aumento del error de Tipo I en comparaciones múltiples como el test-F, pero especifica qué condición es mejor

Esta es una prueba *a posteriori*, en la cual las comparaciones no están planificadas de manera previa, lo que permite corregir las probabilidades infladas de error de Tipo I cuando se hacen comparaciones múltiples. El método Newman-Keuls mantiene el error de Tipo I en α para cada comparación, a diferencia del método HSD, que mantiene el error para el conjunto completo de comparaciones posibles. El test de Newman-Keuls es más potente porque presenta un error de Tipo I más alto para el experimento pero un error de Tipo II más bajo. Por el contrario, HSD es más conservador.

			Población de hipótesis nula	
	Test	Estadístico	Información	Distribución muestral
Una muestra	test-z	media	μ, σ	Normal
Una muestra	test-t	media	μ	distrib-t
Dos muestras correladas	test-t	media de diferencias	$\mu_D = 0$	distrib-t
Dos muestras independientes	test-t	diferencia de medias	$\mu_1 - \mu_2 = 0$	distrib-t
Múltiples muestras*	test-F	varianza	$\mu_1 = \mu_2 = \ldots = \mu_k$	distrib-F
Múlt. muest.* y comparaciones	Newman-Keuls	varianza	$\mu_1 = \mu_2 = \ldots = \mu_k$	distrib-Q

Tabla D.2: Resumen de los test de hipótesis. *$k > 2$ muestras.

Bibliografía

Ahalt, S. C., Krishnamurthy, A. K., Chen, P. y Melton, D. E. (1990) Competitive learning algorithms for vector quantization. *Neural Netw.*, 3(3):277–290. doi: 10.1016/0893-6080(90)90071-R.

Bengio, S., Mariéthoz, J. y Keller, M. (2005) The expected performance curve. En *International Conference on Machine Learning, ICML, Workshop on ROC Analysis in Machine Learning*, Idiap-RR-85-2003. Bonn, Germany.

Bertsimas, D. y Popescu, I. (2005) Optimal inequalities in probability theory: A convex optimization approach. *SIAM J. Optim.*, 15(3):780–804. doi: 10.1137/S1052623401399903.

Bishop, C. M. (1995) *Neural Networks for Pattern Recognition*. Oxford Univ. Press, UK. ISBN 0-19-853864-2.

Bishop, C. M. (2006) *Pattern Recognition and Machine Learning*. Springer, Cambridge, UK. ISBN 978-0387-31073-2.

Boyd, S. y Vandenberghe, L. (2004) *Convex Optimization*. Cambridge Univ. Press, UK. ISBN 978-0-521-83378-3.

Breiman, L. (1996) Bias, variance and arcing classifiers. Technical Report 460, Statistics Department, University of California. http://www.cs.toronto.edu/~delve/data/twonorm/desc.html.

Chang, C.-C. y Lin, C.-J. (2011) LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27. http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

Clark, M. P. (1995) On the resolvability of normally distributed vector parameter estimates. *IEEE Trans. Signal Process.*, 43(12):2975–2981. doi: 10.1109/78.476441.

Doersch, C. (2016) Tutorial on variational autoencoders. *arXiv e-prints*, arXiv:1606.05908.

Duda, R. O., Hart, P. E. y Stork, D. G. (2000) *Pattern Classification*. Wiley–Interscience, New York, NY, 2nd edición. ISBN 978-0471056690.

Fisher, A. (1923) *The Mathematical Theory of Probabilities*. Macmillan, New York, NY.

Fukunaga, K. (1990) *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, CA, 2nd edición. ISBN 978-0-08-047865-4.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. y Bengio, Y. (2014) Generative adversarial networks. *arXiv e-prints*, arXiv:1406.2661.

Haykin, S. (2009) *Neural Networks and Learning Machines*. Pearson-Prentice Hall, New York, NY, 3rd edición. ISBN 978-0-13-147139-9.

Huang, G.-B., Zhu, Q.-Y. y Siew, C.-K. (2006) Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1):489–501. doi: 10.1016/j.neucom.2005.12.126.

Huang, K., Yang, H., King, I., Lyu, M. R. y Chan, L. (2004) The minimum error minimax probability machine. *J. Mach. Learn. Res.*, 5:1253–1286. www.jmlr.org/papers/volume5/huang04a/huang04a.pdf.

Jollife, I. T. (2002) *Principal Component Analysis*. Springer-Verlag, New York, NY, 2nd edición. ISBN 978-0-387-22440-4. doi: 10.1007/b98835.

Lanckriet, G. R., El Ghaoui, L., Bhattacharyya, C. y Jordan, M. I. (2002) A robust minimax approach to classification. *J. Mach. Learn. Res.*, 3:555–582. www.jmlr.org/papers/volume3/lanckriet02a/lanckriet02a.pdf.

LeCun, Y., Cortes, C. y Burges, C. J. (1998) The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist.

Lichman, M. (2013) UCI Machine Learning Repository. http://archive.ics.uci.edu/ml.

Makhzani, A., Shlens, J., Jaitly, N. y Goodfellow, I. J. (2015) Adversarial autoencoders. *arXiv e-prints*, página arXiv:1511.05644.

Marshall, A. W. y Olkin, I. (1960) Multivariate Chebyshev inequalities. *Ann. Math. Statist.*, 31(4):1001–1014. doi: 10.1214/aoms/1177705673.

Martínez-García, J.-A. y Sancho-Gómez, J.-L. (2018) Performance analysis of No-Propagation and ELM algorithms in classification. *Neural Comput. Appl.* doi: 10.1007/s00521-018-3353-0. http://rdcu.be/E68l.

Martínez-García, J.-A., Sancho-Gómez, J.-L., Sánchez-Morales, A. y Figueiras-Vidal, A. R. (2019) Designing non-linear minimax and related discriminants by disjoint tangent configurations applied to RBF networks. *Neurocomputing*. Unpublished. In revision.

Pagano, R. R. (2010) *Understanding Statistics in the Behavioral Sciences*. Wadsworth Cengage Learning, Belmont, CA, 9th edición. ISBN 978-0-495-59652-3.

Peterson, D. W. y Mattson, R. L. (1966) A method of finding linear discriminant functions for a class of performance criteria. *IEEE Trans. Inf. Theory*, 12(3):380–387. doi: 10.1109/TIT.1966.1053913.

Sánchez-Morales, A., Sancho-Gómez, J.-L., Martínez-García, J.-A. y Figueiras-Vidal, A. R. (2019) Improving deep learning performance with missing values via deletion and compensation. *Neural Comput. Appl.* doi: 10.1007/s00521-019-04013-2.

Sancho-Gómez, J.-L., Martínez-García, J.-A., Ahalt, S. C. y Figueiras-Vidal, A. R. (2018) Linear discriminants described by disjoint tangent configurations. *Neurocomputing*, 316:345–356. doi: 10.1016/j.neucom.2018.08.010.

Schölkopf, B. H. y Smola, A. J. (2001) *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA. ISBN 978-0262194754.

Sebastiani, F. (2002) Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47. doi: 10.1145/505282.505283.

Vapnik, V. (1995) *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, NY. ISBN 978-1475732641.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. y Manzagol, P.-A. (2010) Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408. http://dl.acm.org/citation.cfm?id=1756006.1953039.

Widrow, B., Greenblatt, A., Kim, Y. y Park, D. (2013) The No-Prop algorithm: A new learning algorithm for multilayer neural networks. *Neural Netw.*, 37:182–188. doi: 10.1016/j.neunet.2012.09.020.

Índice de Figuras

Índice de Tablas

POLYTECHNIC UNIVERSITY OF CARTAGENA

DEPARTMENT OF INFORMATION TECHNOLOGIES AND COMMUNICATIONS

Machine Learning based on Disjoint Tangent Configurations

DOCTORAL THESIS

Information Technologies and Communications Program

Author: Juan Antonio Martínez García
juan.antonio.mtnez@gmail.com

Director: Dr. José Luis Sancho Gómez
josel.sancho@upct.es

Cartagena (Spain), 2019

Foreword

Why study a doctorate?

The practical reason is to validate the beginning of a professional career related to research, in many cases within teaching, or perhaps in one of the growing number of companies that require a doctorate to carry out R&D. It is also a great learning opportunity, a practical specialization in some field in which some innovative contribution is also made. However, one of the main attractions is that such learning takes place in situations other than those of a class or a job, in which there are fixed guidelines about the content or a selection criteria based on profitability. The Ph.D. student enjoys a freedom that allows them to explore and make decisions about the direction of their work. It is a perfect environment for those who enjoy learning, being guided by an expert in the field, the thesis director, from who their way of learning is assimilated.

"Dimidium facti, qui coepit, habet; sapere aude, incipe. [He who has begun is half done; dare to know; begin!]"

— Horace. Epist. I, 2

Machine learning

The subject chosen for this thesis, machine learning, is a branch of artificial intelligence that aims for computers to learn without explicitly telling them how to do it, but rather from experience through data provided. Artificial intelligence is a main part of the so-called Fourth Industrial Revolution in which the boundaries between physics, biology and the digital are blurred. The real scope this will have in the transformation of society is unimaginable. The activities that the machines will carry out, some of them currently carried out by us, after the initial fear related to the loss of jobs, will raise the question of which capabilities are intrinsically human, once it is found that skills we thought were the property of humans cease to be.

"We are the universe contemplating itself"

— Carl Sagan

Acknowledgments

First of all, I want to thank my parents for all that I am. They worked very hard so that I had all the opportunities that allowed me to develop as a person in all its facets. I appreciate it even more, because they started from more adverse conditions. My only merit has been to take advantage of some of the opportunities I have had. From my father, also an engineer, I learned the love for knowledge

and science, honesty and goodness; I still remember how as a child I preferred him to explain to me the way things work while falling asleep, instead of telling me tales. He died during the completion of the final degree project but I am sure that in life he already conceived what is happening now and what is to come. From my mother, with artistic training, I have learned and continue learning everything that cannot be measured and that constitutes the main axis of life, love, tireless effort, self-demand, aesthetic sensibility and creativity.

I would also like to thank Dr. José Luis Sancho Gómez, Tenured Professor at the Polytechnic University of Cartagena and Principal Investigator of the R&G group of Data Processing and Machine Learning (TDAM), director of the thesis, for the opportunity to have completed the doctorate based on the study of situations of disjoint tangency that he had initiated years ago. I want to highlight his generosity both in the transmission of knowledge and in his dedication and practical advice.

I want to give a special mention to Dr. Aníbal R. Figueiras Vidal, University Professor at the Carlos III University of Madrid and researcher of international prestige, for the generosity of his help in the realization of the scientific articles that have served to provide content to this thesis.

I also remember my school teachers from San Pablo C.E.U. of Murcia, for teaching me the value of knowledge, rigorous work and critical judgment.

Abstract

The aim of a classifier is to decide, with the least possible error, which class a sample or pattern belongs to. In this thesis, a new interpretation of linear discriminants is presented, in which they are described in terms of Disjoint Tangent Configurations (DTC) established between the ellipsoidal probability-level surfaces resulting from the characterization of the data class distributions by their two first moments, the means and the covariance matrices. This is a common framework that allows the design and analysis of several well-known discriminants through an analytical correspondence with other methods: the parametric method, consisting of the minimization of an error function in a one-dimensional projected space in order to determine the parameters of the mathematical expression of the discriminants, *e.g.*, the Fisher, Scatter-based or the Bayes linear discriminant, whose explicit expression is sill unknown; and the minimax convex optimization method, consisting of bounding and minimizing the misclassification probability, *e.g.*, the solution of the Minimax Probabilistic Decision Hyperplane (MPDH) provided by the Minimax Probability Machine (MPM), that minimizes the worst case or maximum risk over all possible distributions characterized with the same first two moments, being suitable when the data class distributions are unknown or do not reflect the actual prior probabilities. It also allows the design of new discriminants, such as a complete Fisher discriminant with an independent term or Quasi-Bayes, which is a geometrical approximation to optimal Bayes with similar accuracy and lower computational cost, a general DTC advantage since it is a non-iterative method.

Classification

DTC

Analytical correspondence

Parametric method

Minimax convex optimization method

Minimax

New discriminants

Low computational cost

In the second part of the thesis, the non-linear versions of the DTC linear discriminants are built using Radial Basis Function Networks (RBFNs) with pre-trained Gaussian kernels by vector quantization techniques. Thus, the input data space is transformed into a higher space with higher linear separability in which is solved the classification problem with a DTC linear discriminant. The resulting non-linear DTC discriminant maintains the properties of the original DTC linear discriminant and allows to solve more complex problems with classes that are not linearly separable.

Non-linear DTC

RBFNs

Pre-trained Gaussian kernels

Experiments demonstrate that the DTC approach leads to good accuracy results with a competitive computational cost in terms of training time, because it is a non iterative solution without training parameters that need to be adjusted, and memory requirements, compared to globally-trained kernel networks.

DTC: good accuracy and competitive computatinal cost

Publications

- Martínez-García, J.-A. and Sancho-Gómez, J.-L. (2018) Performance analysis of No-Propagation and ELM algorithms in classification. *Neural Comput. Appl.* doi: 10.1007/s00521-018-3353-0. http://rdcu.be/E68l
- Sancho-Gómez, J.-L., Martínez-García, J.-A., Ahalt, S. C. and Figueiras-Vidal, A. R. (2018) Linear discriminants described by disjoint tangent configurations. *Neurocomputing*, 316:345–356. doi: 10.1016/j.neucom.2018.08.010
- Sánchez-Morales, A., Sancho-Gómez, J.-L., Martínez-García, J.-A. and Figueiras-Vidal, A. R. (2019) Improving deep learning performance with missing values via deletion and compensation. *Neural Comput. Appl.* doi: 10.1007/s00521-019-04013-2
- Martínez-García, J.-A., Sancho-Gómez, J.-L., Sánchez-Morales, A. and Figueiras-Vidal, A. R. (2019) Designing non-linear minimax and related discriminants by disjoint tangent configurations applied to RBF networks. *Neurocomputing*. Unpublished. In revision

NEUROCOMPUTING

Impact Factor (2018): 4.072

JCR© Category	Rank in Category	Quartile in Category
Computer Science, Artificial Intelligence	**28 of 133**	**Q1**

Publisher: ELSEVIER SCIENCE BV,
PO BOX 211, 1000 AE Amsterdam, Netherlands
ISSN: 0925-2312 **eISSN:** 1872-8286
Research Domain: Computer Science

NEURAL COMPUTING & APPLICATIONS

Impact Factor (2018): 4.664

JCR© Category	Rank in Category	Quartile in Category
Computer Science, Artificial Intelligence	**21 of 133**	**Q1**

Publisher: SPRINGER LONDON LTD,
236 Grays Inn RD, 6th Floor, London WC1X 8HL, England
ISSN: 0941-0643 **eISSN:** 1433-3058
Research Domain: Computer Science

Index

Nomenclature

The next list describes the symbols, acronyms and abbreviations used in the document. Column vectors are in bold type and lower case, matrices are in upper case and scalar values in lower case, the latter two both in plain text.

Algorithms

CV	Cross-Validation
DTC	Disjoint Tangent Configurations
FSCL	Frequency Sensitive Competitive Learning
MEMPM	Minimum Error Minimax Probability Machine
MLP	Multilayer Perceptron
MPM	Minimax Probability Machine
SOCP	Second-Order Cone Program
SVM	Support Vector Machine
TM	Theoretical Method

Iterators

i	i-th sample, i=1, 2, . . . , N
j	j-th class or j-th output node, j=1, 2, . . . , $\tilde{n}$, (except in binary classification, j=1, 2, where there is only one output node, $\tilde{n} = 1$)
k	k-th hidden node, k=1, 2, . . . , m

Input data (samples)

$\mathbf{x}$	Input data, sample (random variable), $\mathbf{x} = [x_1, x_2, \ldots, x_n]$
$\mathbf{x}^{(i)}$	i-th Sample, $\mathbf{x}^{(i)} = [x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}]$
N	Number of samples
n	Dimension or number of variables or features of the samples, number of input nodes
C_j	j-th class
P_j	A priori probability of the j-th class
$\mathbf{m}_j$	Mean vector of the samples of the j-th class
Σ_j	Convariance matrix of the samples of the j-th class

Linear discriminants

$\mathbf{w}$	Weight vector
$\mathbf{x}_0$	Bias vector
ω_0	Threshold value
$\mathcal{H}_{\mathrm{L}}$	Hyper-plane, linear decision boundary
h	Projection function of the samples in the direction $\mathbf{w}$ or one-dimensional projected space
μ	Mean of the projected samples in h
σ^2	Variance of the projected samples in h
f	Class-separability criterion function
s	Parametric method optimization parameter

DTC discriminant

$\mathbf{t}$	Tangent point between the level surfaces and the DTC linear discriminant
α	Relation between the gradients of the level surfaces in the tangent point which determines a discriminant

MPDH problem

MPDH	Minimax Probabilistic Decision Hyperplane
ε	Lower bound of the probability of correct classification

Non-linear discriminants

m	Number of hidden nodes
$\tilde{n}$	Number of output nodes, $\tilde{n} = 1$ in binary classification
$\mathcal{H}_{\mathrm{NL}}$	Non-linear decision boundary
ϕ	Non-linear transformation function
κ	Kernel function
L	Gram matrix
SLFN	Single-hidden Layer Feedforward Neural Network
RBFN	Radial Basis Function Networks

Funtions and operators

E	Mathematical expectation
∂	Partial derivative
exp	Exponential function
P	Probability
pdf	Probability density function
$D_M(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$	Mahalanobis distance between the points $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$
$\mathcal{O}$	Computational cost order

Introduction

Machines, like humans, have the ability to observe a portion of reality (samples) and, like humans again, using their memory (learning weights), make an assumption about reality (inference).

The decision making, of which our lives are full, has its equivalence in the classification problems. In the most simple case, the decision is choosing which situation (class), between two of them, an event belongs to (binary classification). Analytically, the aim is to determine which posterior probability is greater —$P(C_j|\mathbf{x})$, j=1, 2, given an event $\mathbf{x}$, is the probability of belonging to class C_j—. Thus, the decision rule is the ratio of the posterior probabilities in relation to a risk or cost policy that allows to quantitatively establish the importance of being right or wrong in the decision

$$\frac{P(C_2|\mathbf{x})}{P(C_1|\mathbf{x})} \underset{C_1}{\overset{C_2}{\gtrless}} \frac{q_{21} - q_{11}}{q_{12} - q_{22}} \,, \qquad \text{Minimum risk decision rule}$$

q_{dj} being the cost of making the decision that the sample $\mathbf{x}$ belongs to class C_d when the actual situation is it belongs to class C_j.

Theoretical problem

In a theoretical problem in which the joint distributions are known *i.e.*, the probability density functions or likelihoods $P(\mathbf{x}|C_j)$ and the prior probabilities of the classes $P(C_j)$ are known, the optimal solution is Bayesian, that minimizes the decision error and allows to calculate the ratio of posteriors from the likelihoods and the prior probabilities. The decision rule resulting in

$$\frac{P(C_2|\mathbf{x})}{P(C_1|\mathbf{x})} = \frac{P(\mathbf{x}|C_2)P(C_2)}{P(\mathbf{x}|C_1)P(C_1)} \underset{C_1}{\overset{C_2}{\gtrless}} \frac{q_{21} - q_{11}}{q_{12} - q_{22}} \,. \qquad \text{Optimal Bayes decision rule}$$

Real problems

In real problems, the class models $P(\mathbf{x}|C_j)$ are unknown and there are not infinite independent data samples to estimate them. Therefore, without a complete prior information about the classes, it is not possible to achieve the minimum decision cost of the optimal Bayes' solution. Instead, if labeled samples are available, the joint distributions $P(\mathbf{x}, C_j)$ can be estimated from the samples in two different supervised learning approaches: the generative models and the discriminative models. Since sometimes the labeled samples are costly to obtain or the number of classes may be unknown, the unsupervised learning can be used independently or as a first stage of classification. Clustering is the primary use of unsupervised methods.

Generative models

The generative approach estimates the actual joint distribution of the classes –the likelihoods $P(\mathbf{x}|C_j)$ and the priors C_j– assuming the form of the class distributions and estimating its parameters from the samples. Thus, the posterior probabilities are calculated by means of Bayes' theorem. These methods are focused in the representation of the data classes and evaluate the similarity of new samples with respect to the model. The parametric methods, graphs and Markov models are examples of this approach. They present limited performance and the risk of choosing a different model from the actual model.

Discriminative models

On the other hand, the discriminative approach directly estimate the posterior probabilities $P(C_j|\mathbf{x})$ from samples and models the decision border between the classes by means of a surrogate cost that is minimized. The classifier output makes predictions based on the differences between classes. This is the case of logistic regression, support vector machines or neural networks. In these methods, the intermediate variable values of the structure are hard to interpret. They provide high performance although at risk of over-fitting.

Discriminant analysis

Among discriminative models, the discriminant analysis is an effort to find a less computationally costly alternative to the Bayes discriminant. Instead of the estimation of the parameters of the density functions like generative models the mathematical model of the decision border (discriminant) is established and its parameters are estimated from samples, like the parametric discriminants of Chapter 1.

Neural networks

Neural networks, within the discriminative models, are suitable to solve more complex problems that need non-linear decision boundaries. They have universal approach capabilities. They can be trained in a supervised way or in the mixing of two separated stages, unsupervised first and supervised last, being the training complexity of each stage less than that of joint training, providing better training speed and lower computational cost. In this form of stage training, the supervised stage can be trained using the discriminant analysis, as shown in Chapter 3.

DTC (discriminant analysis and neural networks)

The linear discriminant analysis from disjoint tangent configurations (DTC), Chapter 2, is based on the geometric interpretation that derives from the characterization of the data classes by their first two moments, which originates probability level curves that are ellipsoids, where the discriminants correspond to the tangent hyperplanes to the different ellipsoids. DTC belongs to the discriminant analysis and is used as a design and interpretation framework for both known discriminants –such as those of the parametric method and those of the minimax convex-optimization method, with which it is possible to establish an analytical correspondence– as well as new discriminants –as the one obtained from the approximation to the optimal Bayes, with similar precision and a lower computational cost since DTC is a non-iterative method–. By means of the use of single-hidden layer neural networks with Gaussian kernels, Chapter 3, it is possible to construct non-linear DTC discriminants from linear DTCs with a lower computational cost due to the possibility of pre-training the Gaussian kernels in an unsupervised way. From the geometric interpretation of the DTC, the minimax solution is naturally derived, both in linear and non-linear discriminants, suitable for problems in which it is desired to minimize the risk associated with distributions of the data classes or prior probabilities which are unknown.

Part I

Linear DTC discriminants

Given a set of samples or patterns which belong to two classes (binary problem), there is a need to decide (**classification**) which class new samples belong to, making the least possible error. Linear discriminants are possible classifiers, *i.e.*, a hyperplane in the generic case of multiple dimensions which separates space in two regions, one per class, allowing to decide which of them a new sample belongs to. Although linear discriminants are only optimal for normally distributed classes with equal covariance matrices, in many cases, they are preferred for simplicity, robustness, and ease of interpretation.

The **Bayes'** classifier is optimal because it minimizes the probability of error (Fukunaga, 1990), but requires the knowledge of the probability density functions of the classes from estimation techniques which are computationally complex, needing large amounts of data to provide accurate results.

Thus, simpler procedures like the **parametric techniques** have been developed, they specify the mathematical form of the classifier followed by its parameter estimation. Such is the case of the Fukunaga (1990) procedure of linear discriminant design for binary problems. This method is optimal with respect to a **separability criterion** defined in a one-dimensional projected space, *i.e.*, a direction along which the projected data of one class are maximally separated from the projected data of the other. Different linear discriminants are obtained from different separability criteria. The most important criteria are the Bayes error for normal distributions, the Fisher (1923) criterion and other criteria based on scatter matrices (Duda et al., 2000).

Finally, a new interpretation of linear discriminants is presented in which they are described in terms of Disjoint Tangent Configurations (**DTC**) by Sancho-Gómez, Martínez-García, Ahalt and Figueiras-Vidal (2018), whose decision boundaries are the tangent hyperplanes to the different probability-level surfaces (ellipsoids) defined by the first two moments of the distributions of the classes. It is possible to establish an **analytical correspondence** between the formulation of the parametric method and the interpretation of its discriminants as disjoint tangent configurations (DTC). It is also possible to establish a analytical correspondence with the discriminants obtained by the convex optimization method (Bertsimas and Popescu, 2005) based on minimizing the maximum error probability, such as the Minimax Probability Machine by Lanckriet et al. (2002), which is the solution of the Minimax Probabilistic Decision Hyperplane problem, and the Minimum Error Minimax Probability Machine by Huang et al. (2004), which provides a Bayesian solution. DTC provides a **direct solution** with a competitive computational cost in terms of speed and memory, in contrast with the iterative convex optimization method. This new design framework also allows obtaining **new discriminants** with very interesting properties, in particular, the Quasi-Bayes discriminant, which provides a very close accuracy to that of the Bayes Linear Discriminant with the advantage of needing a lower computational cost.

1 Design of linear discriminants

1.1 *Parametric design*

The parametric method of Fukunaga (1990) is a simpler procedure for designing binary linear classifiers in which the analytical form of the classifier is specified including a finite set of free parameters that need to be fitted in an optimization process, *i.e.*, the weight and bias vectors of the linear discriminant function $h : \mathbb{R}^n \to \mathbb{R}$, expressed as follows

$$h(\mathbf{x}) = \mathbf{w}^T(\mathbf{x} - \mathbf{x}_0) \,, \tag{1.1}$$

where, $\mathbf{x} \in \mathbb{R}^n$ are the input samples, $\mathbf{w} \in \mathbb{R}^n$ the weight vector and $\mathbf{x}_0 \in \mathbb{R}^n$ the bias vector and any point of the discriminant.

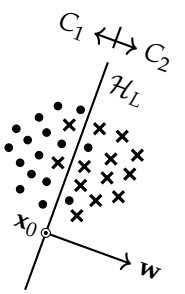


Figure 1.1: Linear discriminant example with samples from class C_1 (points) and class C_2 (crosses).

Thus, the decision rule of a two-class classification problem is

DECISION RULE

$$h(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + \omega_0 \underset{C_2}{\overset{C_1}{\gtrless}} 0 \,, \tag{1.2}$$

where

$$\omega_0 = -\mathbf{w}^T\mathbf{x}_0 \,, \tag{1.3}$$

and it is interpreted as the projection of the samples onto the direction of the vector $\mathbf{w}$, which are classified as belonging to either class C_1 or class C_2 depending on whether variable $z = \mathbf{w}^T\mathbf{x}$ is greater or less than $-\omega_0$, called the threshold value. Thus, $h(\mathbf{x})$ is also called the one-dimensional projection space.

Equation $h(\mathbf{x}) = 0$ describes the decision boundary, which is in the general n-dimensional case the hyperplane $\mathcal{H}_L(\mathbf{w}, \omega_0) = \{\mathbf{x} \in \mathbb{R}^n \,|\, \mathbf{w}^T\mathbf{x} + \omega_0 = 0\}$ in which the weight vector $\mathbf{w}$ determines its orientation and the bias vector $\mathbf{x}_0$ fixes its relative position in the data space, see Figure 1.1. Thus, a linear discriminant, by means of a hyperplane, divides this space into two half-spaces corresponding to the decision regions.

DECISION BOUNDARY

The designing of a linear classifier consists of finding the optimum weight vector $\mathbf{w}$ and the threshold value ω_0 (or the bias vector $\mathbf{x}_0$) that provide the smallest classification error in the one-dimensional projected h-space as a result of the optimization of a selected class-separability criterion. That is to say, first, choosing a separability

DESIGN. Finding $\mathbf{w}$ and ω_0 that present less classification error under the f separability criterion.

criterion by means of the function f that measures the degree of separation of the classes, and then finding the projection direction $\mathbf{w}$ that, according to the separability criterion chosen, has the lowest classification error. Figure 1.2 shows how the projection direction $\mathbf{w}$ presents a lower classification error (shaded area) than the direction $\mathbf{w}'$ in two classes represented by their means and covariance matrices.

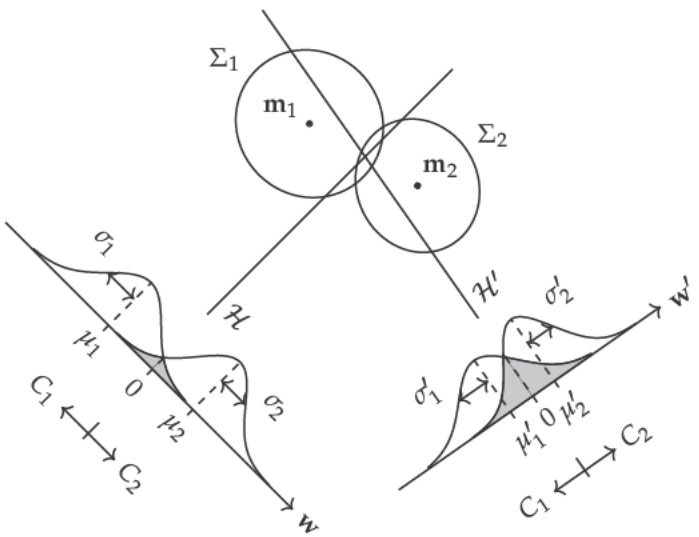


Figure 1.2: Projected space h onto two different directions, $\mathbf{w}$ and $\mathbf{w}'$, of the linear discriminants.

When $\mathbf{x}$ is normally distributed or n is large, $h(\mathbf{x})$ is also normal or close to normal, due to the central limit theorem, respectively. In this case, the appropriate criterion f to measure the class separability depends on the means and variances of $h(\mathbf{x})$, *i.e.*, $f(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2)$, which are

SEPARABILITY CRITERION. Is a function of the means and variances in the projected space.

$$\begin{aligned}\mu_j &= E\{h(\mathbf{x})|C_j\} = \mathbf{w}^T E\{\mathbf{x} \mid C_j\} + \omega_0 \\ &= \mathbf{w}^T \mathbf{m}_j + \omega_0 \,, \end{aligned} \tag{1.4a}$$

$$\begin{aligned}\sigma_j^2 &= Var\{h(\mathbf{x})|C_j\} = \mathbf{w}^T E\{(\mathbf{x} - \mathbf{m}_j)(\mathbf{x} - \mathbf{m}_j)^T \mid C_j\}\,\mathbf{w} \\ &= \mathbf{w}^T \Sigma_j \mathbf{w} \,, \end{aligned} \tag{1.4b}$$

CENTRAL LIMIT THEOREM: the means from different samples of a distribution approaches a normal distribution as the sample size gets larger.

Thus, the scalar product of a pattern by the weight vector, $z_i = \mathbf{w}^T\mathbf{x}^{(i)} = w_1 x_1^{(i)} + w_2 x_2^{(i)} + \ldots + w_n x_n^{(i)}$, $i=1,2,\ldots,N$, can be considered a "weighted mean" of the components of $\mathbf{x}^{(i)}$, being the projected patterns $\{z_1, z_2, \ldots, z_N\}$, approximately a normal distribution if n is large enough.

where $\mathbf{m}_j \in \mathbb{R}^n$ and $\Sigma_j \in \mathbb{R}^{n\times n}$, j=1,2, are respectively the mean vector and covariance matrix of the samples of each class before projecting. As mentioned before, the optimal values of the parameters $\mathbf{w}$ and ω_0 are obtained from the optimization of the separability criterion function f (Fukunaga, 1990), resulting in

$$\mathbf{w}^{\mathrm{P}} = [s\Sigma_1 + (1-s)\Sigma_2]^{-1}\,(\mathbf{m}_2 - \mathbf{m}_1) \,, \tag{1.5}$$

PARAMETRIC METHOD $(.)^{\mathrm{P}}$
Separab. crit. $f(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2)$
GENERAL SOLUTION:
$\mathbf{w}^{\mathrm{P}}(s)$ (1.5), $\omega_0^{\mathrm{P}}(s)$ (1.7)

where

$$s = \frac{\partial f/\partial\sigma_1^2}{\partial f/\partial\sigma_1^2 + \partial f/\partial\sigma_2^2} \,, \tag{1.6}$$

and ω_0^{P}, which is obtained as the solution of

$$\frac{\partial f}{\partial \mu_1} + \frac{\partial f}{\partial \mu_2} = 0 \,. \tag{1.7}$$

Notice that $\mathbf{w}^{\mathrm{P}}$ (1.5) does not depend on the selection of f, which only affects s (1.6). Moreover, this solution is valid not only for normal class conditional distributions, but also for any binary classification problem in which the means and the covariance matrices are known.

The different linear discriminants correspond to different selections of f, some of them are presented below.

1.1.1 *Bayes discriminant for Gaussian distributions*

As mentioned before, in the case of normal distributions ($\mathbf{x} \sim N(\mathbf{m}_j, \Sigma_j)$), $h(\mathbf{x})$ is also normal, and therefore the classification error in the h-space is the Bayes error. The Bayes linear discriminant aims to minimize the classification error, *i.e.*, the separability criterion is the Bayes error. The search of the minimum error can be done through an iterative procedure described later.

To determine the discriminant parameters, the Bayes error can be expressed as a function of μ_j and σ_j^2 as

$$f = \frac{P_1}{\sqrt{2\pi}} \int_{-\frac{\mu_1}{\sigma_1}}^{+\infty} \exp\left(\frac{-\xi^2}{2}\right) d\xi + \frac{P_2}{\sqrt{2\pi}} \int_{-\infty}^{-\frac{\mu_2}{\sigma_2}} \exp\left(\frac{-\xi^2}{2}\right) d\xi \,, \quad (1.8)$$

where P_j, j=1, 2, is the prior probability of the j-th class.

It can be proven (Fukunaga, 1990) that, in this case, the optimum linear discriminant is given by $\mathbf{w}^{\mathrm{P}}$ (1.5) with

$$s = \frac{-\mu_1/\sigma_1^2}{-\mu_1/\sigma_1^2 + \mu_2/\sigma_2^2} \,, \quad (1.9)$$

with $0 < s < 1$ because $\mu_1 < 0$ and $\mu_2 > 0$. On the other hand, ω_0^{PB} is obtained as the solution of (1.7), which results in the following equation

$$\frac{P_1}{\sigma_1\sqrt{2\pi}} \exp\left(\frac{-\mu_1^2}{2\sigma_1^2}\right) = \frac{P_2}{\sigma_2\sqrt{2\pi}} \exp\left(\frac{-\mu_2^2}{2\sigma_2^2}\right) . \quad (1.10)$$

PARAMETRIC METHOD
BAYES LINEAR DISC. $(.)^{\mathrm{PB}}$
f: Bayes error (1.8)
SOLUTION (not explicit):
$\mathbf{w}^{\mathrm{P}}(s^*)$ (1.5)
$\omega_0^{\mathrm{P}}(s^*)$ (1.7) $\rightarrow$ $\omega_0^{\mathrm{PB}}(s^*)$ (1.11)
s^*: optimum (minimum Bayes error)

The expression for ω_0^{PB} as a function of s and $\mathbf{w}^{\mathrm{PB}}$ is easily obtained substituting μ_1 and μ_2 of (1.4a) into (1.9) and isolating ω_0^{PB}

$$\omega_0^{\mathrm{PB}} = -\frac{s\sigma_1^2(\mathbf{w}^{\mathrm{PB}})^T\mathbf{m}_2 + (1-s)\sigma_2^2(\mathbf{w}^{\mathrm{PB}})^T\mathbf{m}_1}{s\sigma_1^2 + (1-s)\sigma_2^2} \,. \quad (1.11)$$

Notice that s in (1.9) is a function of $\mathbf{w}^{\mathrm{P}}$ and ω_0^{P} from (1.4), and $\mathbf{w}^{\mathrm{P}}$ in (1.5) is a function of s; therefore, an explicit optimum solution for $\mathbf{w}^{\mathrm{PB}}$ and ω_0^{PB} cannot be obtained this way because of their interdependence. Nevertheless there exists an alternative procedure to find the Bayes linear discriminant called the Theoretical Method (TM) (Fukunaga, 1990; Peterson and Mattson, 1966) shown below

Algorithm 1: Theoretical Method (TM)

Data: P_j, $\mathbf{m}_j$, Σ_j, j=1, 2
Result: $\mathbf{w}^*$, ω_0^*
for *(swept of* $s \in [0,1]$ *in* Δs *steps)* **do**
  $\mathbf{w} \leftarrow s$ in (1.5)
  $\sigma_j^2 \leftarrow \mathbf{w}$ in (1.4b)
  $\omega_0 \leftarrow \mathbf{w}$ and σ_j^2 in (1.11)
  $\mu_j \leftarrow \mathbf{w}$ and ω_0 in (1.4a)
  $f \leftarrow \sigma_j^2$ and μ_j in (1.8)
  if *f is minimum* **then**
    $\mathbf{w}^* \leftarrow \mathbf{w}$
    $\omega_0^* \leftarrow \omega_0$

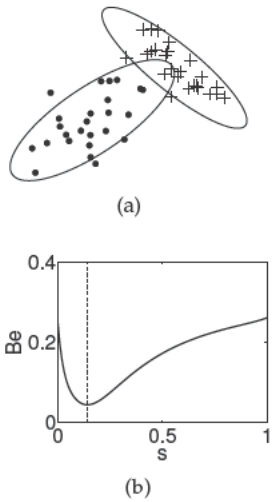


Figure 1.3: (a) Level curves corresponding to two-dimensional normal populations. (b) Simulation results of the Theoretical Method (Bayes error (Be) vs. s and the minimum value in s^*).

Although this is an efficient and easy process, it is an iterative procedure whose result depends on the step Δs. Figure 1.3 illustrates this, it is observed how Δs has to be small enough in order to ensure a satisfactory discriminant.

1.1.2 *Fisher discriminant*

The Fisher (1923) separability criterion is given by

$$f = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}\,, \tag{1.12}$$

which measures the difference of the two means normalized by the averaged variance. Taking derivatives of f with respect to σ^2

$$\frac{\partial f}{\partial \sigma_1^2} = \frac{\partial f}{\partial \sigma_2^2} = \frac{-(\mu_1 - \mu_2)^2}{(\sigma_1^2 + \sigma_2^2)^2}\,, \tag{1.13}$$

and substituting in (1.6) results in $s = 0.5$. Thus, the optimum $\mathbf{w}$ from (1.5) is as follows

PARAMETRIC METHOD
FISHER DISC. $(.)^{\mathrm{PF}}$
f (1.12)
SOLUTION:
$\mathbf{w}^{\mathrm{P}}(s=0.5)$ (1.5) $\rightarrow \mathbf{w}^{\mathrm{PF}}$ (1.14)
without ω_0

$$\mathbf{w}^{\mathrm{PF}} = \left[\frac{1}{2}\Sigma_1 + \frac{1}{2}\Sigma_2\right]^{-1}(\mathbf{m}_2 - \mathbf{m}_1)\,. \tag{1.14}$$

This separability criterion does not depend on ω_0 because the subtraction of μ_2 from μ_1 in (1.12) from (1.4a) eliminates ω_0, so that it cannot be calculated from maximizing f. Hence, the Fisher solution is not a complete discriminant but only an optimum projection direction. Of course, there are complementary procedures to determine ω_0^{PF}, such as using the equivalence with a normal distribution or maximizing the separation of the projected values of the samples.

1.1.3 *Scatter-based discriminant*

Another important criterion for class separability is

$$f = \frac{P_1\mu_1^2 + P_2\mu_2^2}{P_1\sigma_1^2 + P_2\sigma_2^2} \,, \tag{1.15}$$

which measures the between-class scatter (around zero) normalized by the within-class scatter (Fukunaga, 1990). Taking partial derivatives of f with respect to σ_1^2 and σ_2^2

$$\frac{\partial f}{\partial \sigma_i^2} = \frac{-P_i(P_1\mu_1^2 + P_2\mu_2^2)}{(P_1\sigma_1^2 + P_2\sigma_2^2)^2} \,, \tag{1.16}$$

and substituting in s (1.6) results in $s = P_1$. Thus, the optimum $\mathbf{w}^{\mathrm{P}}$ from (1.5) is as follows

$$\mathbf{w}^{\mathrm{PS}} = [P_1\Sigma_1 + P_2\Sigma_2]^{-1}\,(\mathbf{m}_2 - \mathbf{m}_1) \,. \tag{1.17}$$

PARAMETRIC METHOD
SCATTER DISC. $(.)^{\mathrm{PS}}$
f (1.15)
SOLUTION:
$\mathbf{w}^{\mathrm{P}}(s=P_1)$ (1.5) $\rightarrow \mathbf{w}^{\mathrm{PS}}$ (1.17)
$\omega_0^{\mathrm{P}}(s=P_1)$ (1.7) $\rightarrow \omega_0^{\mathrm{PS}}$ (1.19)

ω_0^{P} is obtained taking partial derivatives of f with respect to μ_1^2 and μ_2^2

$$\frac{\partial f}{\partial \mu_i} = \frac{2P_i\mu_i}{P_1\sigma_1^2 + P_2\sigma_2^2} \,, \tag{1.18}$$

and substituting in (1.7) with (1.4a) results in

$$\omega_0^{\mathrm{PS}} = -(\mathbf{w}^{\mathrm{PS}})^T\,[P_1\mathbf{m}_1 + P_2\mathbf{m}_2] \,, \tag{1.19}$$

which shows that if $\mathbf{w}^{\mathrm{PS}}$ is multiplied by a constant, ω_0^{PS} changes by the same factor, and therefore the decision border of the discriminant is the same.

1.2 *Minimax convex-optimization design*

Bound and minimize error

The minimax convex-optimization method makes use of the Marshall and Olkin (1960) inequalities based on the moments of the data class distributions and consists of minimizing the probability of misclassification without making distributional assumptions about the class-conditional densities because it does not offer enough generality and validity. Therefore, the misclassification probability is controlled in a worst-case way, first bounded from the first two moments of the data classes distributions, as shown in Appendix A, and then minimized. Thus, the solution is valid for all possible choices of class-conditional densities with a given mean and covariance matrix.

Hence, the general optimization problem can be expressed as follows

GENERAL MINIMAX OPTIMIZATION PROBLEM

$$\begin{aligned} \max_{\varepsilon_1,\varepsilon_2,\mathbf{w}\neq\mathbf{0},\omega_0} \quad & \lambda_1\varepsilon_1 + \lambda_2\varepsilon_2 \quad s.t. \\ & \inf_{\mathbf{x}\sim(\mathbf{m}_1,\Sigma_1)} P\{\mathbf{w}^T\mathbf{x} + \omega_0 \geq 0\} \geq \varepsilon_1 \\ & \inf_{\mathbf{x}\sim(\mathbf{m}_2,\Sigma_2)} P\{\mathbf{w}^T\mathbf{x} + \omega_0 \leq 0\} \geq \varepsilon_2 \,, \end{aligned} \tag{1.20}$$

where $\mathbf{x} \sim (\mathbf{m}_j, \Sigma_j)$ represents all arbitrary distributions with the same mean $\mathbf{m}_j$ and covariance Σ_j; $\lambda_j \in \mathbb{R}$ and $\varepsilon_j \in \mathbb{R}$ being constants, ε_j bounds the probability of correct classification over all the distributions described. For simplification purposes, let us consider the case of equal ε for both classes and, therefore, without the need for the λ_j constants. In order to minimize the maximum classification error, the lower bound ε of the probability of success of the discriminant is maximized, varying ε itself and the discriminant parameters ($\mathbf{w}$, ω_0), restrained to the condition that the infimum of the correct classification probabilities of the distributions with share the same mean $\mathbf{m}_j$ and covariance matrix Σ_j is greater or equal than ε.

Therefore, by maximizing the minimum probability of success ε, due to the duality principle, the maximum probability of error $(1-\varepsilon)$ is minimized –minimization of the worst case–, being the probability of success inferiorly bounded and superiorly the probability of error. Since the classification error depends on the data distribution and this solution is valid for all arbitrary distributions with the same first two moments, minimizing the maximum risk means finding the discriminant which minimizes the error produced by the distribution with the biggest error, what defines the minimax criterion.

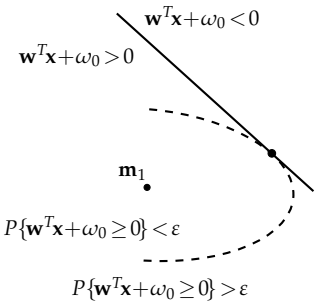


Figure 1.4: Bounding of the correct classification probability in the optimization method.

Minimax solution

The minimax solution, Appendix B, is an important issue in cases such as when the number of training data of each class does not reflect the actual prior probabilities. Therefore, minimax is a natural classification criterion in the absence of prior information of the true frequency of the two classes. For this reason, many researchers prefer to use classifiers operating at Equal Error Rate (EER), that is, classifiers that minimize the maximum of the false alarm and miss rates (Sebastiani, 2002; Bengio et al., 2005). Moreover, the minimax problem can also be addressed when the information of the class distributions is unknown or not exact. Note that (1.20) uses bounds, and consequently its solution does not have to satisfy the EER condition, *i.e.*, it is an approximate solution.

Note that ε is also a separability criterion f, even if it does not have the form $f(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2)$ and, therefore, the solution can be expressed in terms of (1.5)-(1.6). Figure 1.4 shows ε, the lower bound of the success probability (left side of the discriminant), in dashed line. As the value of ε grows, the distance of Mahalanobis d from the mean $\mathbf{m}_1$ increases, which means enlarging the dashed line to the right. The maximum value that ε can reach, or the associated probability-level surface, which is an ellipsoid in the general case, is tangent to the border of the discriminant. The discriminant is common for both classes and is also modified in the process of maximization of ε, otherwise, the discriminant would be closer to one of the two classes and for the other class the value of ε would not be optimal. In the solution, the discriminant is tangent to the two probability ellipses. The ellipses grow at speed $\kappa(\varepsilon) = \sqrt{\frac{\varepsilon}{1-\varepsilon}}$.

Convex optimization. Iterative solution

In Lanckriet et al. (2002) and Huang et al. (2004), the optimization problem (1.20) is addressed by the Second-Order Cone Program (SOCP) by Boyd and Vandenberghe (2004), solving by means of an iterative least-squares approach. It also includes a regularization process of the Hessian matrix to increase the computational stability, and the issue of robustness with respect to the means and covariances estimation errors is addressed via a regularization of the input data.

1.2.1 *Bayes discriminant for Gaussian distributions*

The solution of the Minimum Error Minimax Probability Machine (MEMPM) is provided by Huang et al. (2004) as the solution of the optimization (1.20) by maximizing the success probability of both classes ε_j, with $\lambda_j = P_j$, j=1, 2, as follows

MEMPM (minimax optimization method)

$$\begin{aligned} \max_{\varepsilon_1,\varepsilon_2,\mathbf{w}\neq\mathbf{0},\omega_0} \quad & P_1\varepsilon_1 + (1-P_1)\varepsilon_2 \quad s.t. \\ & \inf_{\mathbf{x}\sim(\mathbf{m}_1,\Sigma_1)} P\{\mathbf{w}^T\mathbf{x}+\omega_0 \geq 0\} \geq \varepsilon_1 \\ & \inf_{\mathbf{x}\sim(\mathbf{m}_2,\Sigma_2)} P\{\mathbf{w}^T\mathbf{x}+\omega_0 \leq 0\} \geq \varepsilon_2 \ . \end{aligned} \tag{1.21}$$

MEMPM minimizes the worst-case (minimax behavior) Bayes error. Thus, MEMPM becomes, under Gaussian conditions, the Bayes optimal discriminant, while in a general case is an approximation.

1.2.2 *MPDH discriminant*

Lanckriet et al. (2002) propose the Minimax Probability Machine (MPM) for binary classification as a solution of the Minimax Probabilistic Decision Hyperplane (MPDH), which is the solution of the optimization problem (1.20) when the success probabilities of each class are equal, $\varepsilon_1 = \varepsilon_2 = \varepsilon$,

MPDH (minimax optimization method)

$$\begin{aligned} \max_{\varepsilon,\mathbf{w}\neq\mathbf{0},\omega_0} \quad \varepsilon \quad s.t. \quad & \inf_{\mathbf{x}\sim(\mathbf{m}_1,\Sigma_1)} P\{\mathbf{w}^T\mathbf{x}+\omega_0 \geq 0\} \geq \varepsilon \\ & \inf_{\mathbf{x}\sim(\mathbf{m}_2,\Sigma_2)} P\{\mathbf{w}^T\mathbf{x}+\omega_0 \leq 0\} \geq \varepsilon \ . \end{aligned} \tag{1.22}$$

As mentioned, the lower bound of the correct classification probability is maximized, for any arbitrary distribution with the same mean and covariance matrix, to minimize the worst case, the maximum misclassification probability.

Lanckriet et al. (2002) prove that the solution parameters ε_* and $\mathbf{w}_*$ of (1.22) are related by the following equation

$$1-\varepsilon_* = \frac{\left(\sqrt{\mathbf{w}_*^T\Sigma_1\mathbf{w}_*} + \sqrt{\mathbf{w}_*^T\Sigma_2\mathbf{w}}\right)^2}{1+\left(\sqrt{\mathbf{w}_*^T\Sigma_1\mathbf{w}_*} + \sqrt{\mathbf{w}_*^T\Sigma_2\mathbf{w}_*}\right)^2} \ , \tag{1.23}$$

ε_* can be obtained from this equation when the optimal MPDH is found by the MPM algorithm or any other optimization procedure.

Note that MPM is a particular case of MEMPM.

2

Disjoint Tangent Configurations discriminants

The method of discriminants based on Disjoint Tangent Configurations (DTC), (Sancho-Gómez et al., 2018), establishes a new interpretation of linear discriminants for binary classification. Like in the optimization convex method (Bertsimas and Popescu, 2005) of MPM (Lanckriet et al., 2002) and MEMPM (Huang et al., 2004), the class distributions are also characterized by their first two moments, the mean vector $\mathbf{m}$ and the covariance matrix Σ, which produces probability level surfaces that are ellipsoids. Thus, the DTC discriminants are defined by the tangent points between different ellipsoids. Figure 2.1 shows two possible tangent points, $\mathbf{t}_1$ and $\mathbf{t}_2$, in two different tangent configurations between ellipsoids –ellipses in the two-dimensional case– and the associated DTC linear discriminants, whose decision boundaries are the tangent lines to the ellipsoids that passes through the tangent points already mentioned. The distributions of data classes are not limited to Gaussian and may be unknown. An estimation of the mean vectors and covariance matrices from the samples is only necessary.

DTC METHOD: tangent ellipsoids

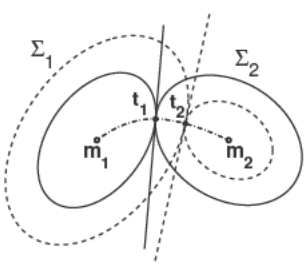


Figure 2.1: Two-dimensional example of two linear DTC discriminants (solid and dashed lines), the tangent points $\mathbf{t}_1$ and $\mathbf{t}_2$ and the curve of all possible tangent points (dashed and dotted curve between means).

This method has three main advantages: First, it is a framework that provides a common interpretation for linear discriminants with an analytical correspondence with the parametric method and the convex optimization method; second, it is not an iterative optimization process, like the optimization method of MPM and MEMPM, but a direct method to calculate the ellipsoids and their tangent points with a competitive computational cost, in terms of speed and memory need; and third, it offers new solutions such as that derived from the geometrical interpretation of the optimum quadratic Bayes discriminant related to DTC, called Quasi-Bayes due to its similar accuracy to Bayes with lower computational cost, or a complete Fisher discriminant, which includes the independent term.

DTC advantages

- Common interpretation framework
- Competitive cost
- Design of new discriminants

2.1 *Analytic expression of the DTC discriminants*

Characterization of the classes by their first two moments. Tangent ellipsoids

To establish the analytical form of the DTC discriminants, let us consider the level-probability surfaces of two normal distributions $p_j(\mathbf{x})$, $j=1,2$, described by the mean vectors $\mathbf{m}_j \in \mathbb{R}^n$ and the covariance matrices $\Sigma_j \in \mathbb{R}^{n\times n}$ which characterize the distributions of the data classes. Many tangent points can be obtained between the different level surfaces of $p(\mathbf{x})$ of each class, which are ellipsoids because they are defined by the first two moments. The decision boundary of each DTC linear discriminant is the hyperplane that passes through the tangent point between a pair of ellipsoids and it is also tangent to them.

A necessary and sufficient condition for a DTC is that the gradient vectors of the class-conditional densities at the tangent point $\mathbf{t}$ are parallel and have opposite orientations,

$$\nabla p_1(\mathbf{t}) = \beta \cdot \nabla p_2(\mathbf{t}) \,, \tag{2.1}$$

where β is a negative real number. Therefore, consistent with the expression of a generic linear discriminant (1.1), the discriminant function associated with a DTC can be expressed as

$$h^{\mathrm{DTC}}(\mathbf{x}) = (\mathbf{w}^{\mathrm{DTC}})^T\mathbf{x} + \omega_0^{\mathrm{DTC}} \,, \tag{2.2}$$

where

$$\mathbf{w}^{\mathrm{DTC}} = \nabla p_j(\mathbf{t}) \,, \tag{2.3a}$$
$$\omega_0^{\mathrm{DTC}} = -(\mathbf{w}^{\mathrm{DTC}})^T\mathbf{t} \,, \tag{2.3b}$$

choosing the tangent point as the bias vector $\mathbf{x}_0^{\mathrm{DTC}} = \mathbf{t}$ (recall that any linear discriminant vector is valid as a bias vector) and the gradient vector of the class-conditional densities at $\mathbf{t}$ as the weight vector $\mathbf{w}$. Figure 2.2 shows the tangent point, the gradient vectors and the linear decision boundary associated with the DTC.

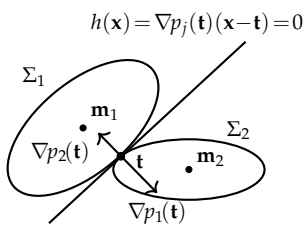


Figure 2.2: Linear discriminant based on a DTC. The decision boundary $h(\mathbf{x}) = 0$, the tangent point $\mathbf{t}$ and the gradient vector of the level surfaces at $\mathbf{t}$, $\nabla p_j(\mathbf{t})$, j=1, 2.

The two types of tangent configurations between two-dimensional ellipses are observed in Figure 2.3. For a fixed level curve of class C_1 (ellipse labeled with A), either a Disjoint Tangent Configuration (DTC) or an Overlapping Tangent Configuration (OTC) can occur with class C_2. In the first case, the level curve of the class C_2 (labeled with B) is tangent and disjoint to the curve A, *i.e.*, the intersection of the areas that enclose each one is null. In the second case, the level curve A is enclosed by the level curve corresponding to C_2 (labeled with B'). Since the aim is to discriminate patterns, only DTC is considered because OTC is not useful.

DTC and OTC tangencies

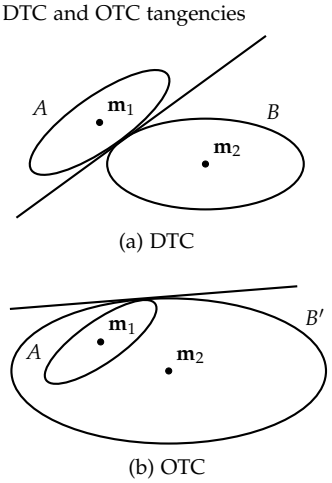


Figure 2.3: Tangency types.

Notice two facts. First, given a level surface of the class C_1, there exists a unique level surface of C_2 that is tangent and disjoint to the former. Second, every DTC determines a unique hyperplane which is tangent to both ellipsoids and passes through their tangent point.

Tangent point calculation

Since to characterize the data classes C_j, j=1, 2, DTC only considers its first two moments, the following normal class-conditional density is used to calculate the DTC tangent point,

$$p_j(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_j|}} \exp\left(-\frac{1}{2} D_M^2(\mathbf{x}, \mathbf{m}_j)\right) , \tag{2.4}$$

where

$$D_M(\mathbf{x}, \mathbf{m}_j) = \sqrt{(\mathbf{x} - \mathbf{m}_j)^T \Sigma_j^{-1} (\mathbf{x} - \mathbf{m}_j)} \tag{2.5}$$

is the Mahalanobis distance from any point $\mathbf{x}$ to the mean $\mathbf{m}_j$ of each class. The level curves of $p_j(\mathbf{x})$ are ellipsoids and are defined by

$$p_j(\mathbf{x}) = c_j , \tag{2.6}$$

c_j being positive real, or equivalently

$$D_M(\mathbf{x}, \mathbf{m}_j) = r_j , \tag{2.7}$$

with $r_j = \sqrt{-\ln\left(c_j^2 (2\pi)^n |\Sigma_j|\right)}$, being $r_j > 0$.

For each value of r_j, (2.7) represents an ellipsoid E_j of dimension n that encloses a probability mass. The probability mass in a region S is defined as the probability of a pattern $\mathbf{x}$ drawn from a distribution $p_j(\mathbf{x})$ falls inside S. Thus,

$$E_j \triangleq \{\mathbf{x} : D_M(\mathbf{x}, \mathbf{m}_j) = r_j\} \tag{2.8}$$

Applying the gradient operator to $p_j(\mathbf{x})$ in (2.4) gives

$$\nabla p_j(\mathbf{x}) = -p_j(\mathbf{x}) \cdot D_M(\mathbf{x}, \mathbf{m}_j) \cdot \nabla D_M(\mathbf{x}, \mathbf{m}_j) . \tag{2.9}$$

Substituting (2.9) in (2.1) in the tangent point $\mathbf{x} = \mathbf{t}$ and considering $\nabla D_M^2(\mathbf{x}, \mathbf{m}_j) = 2 \cdot D_M(\mathbf{x}, \mathbf{m}_j) \cdot \nabla D_M(\mathbf{x}, \mathbf{m}_j)$, the condition for the tangent point given can be also expressed as

Gradients in the tangent point

$\nabla p_1(\mathbf{t}) = \beta \cdot \nabla p_2(\mathbf{t})$

$\nabla D_M^2(\mathbf{t}, \mathbf{m}_1) = \alpha \cdot \nabla D_M^2(\mathbf{t}, \mathbf{m}_2)$

$\nabla D_M(\mathbf{t}, \mathbf{m}_1) = \gamma \cdot \nabla D_M(\mathbf{t}, \mathbf{m}_2)$

$\gamma = \alpha \cdot \frac{D_M(\mathbf{t}, \mathbf{m}_2)}{D_M(\mathbf{t}, \mathbf{m}_1)}$

$$\nabla D_M^2(\mathbf{t}, \mathbf{m}_1) = \alpha \cdot \nabla D_M^2(\mathbf{t}, \mathbf{m}_2) , \tag{2.10}$$

α being real negative constant given by

$$\alpha = \beta \cdot \frac{p_2(\mathbf{t})}{p_1(\mathbf{t})} . \tag{2.11}$$

Applying the gradient operator to $D_M(\mathbf{x}, \mathbf{m}_j)$ (2.5), the following is obtained

$$\nabla D_M(\mathbf{x}, \mathbf{m}_j) = \frac{1}{D_M(\mathbf{x}, \mathbf{m}_j)} \cdot \Sigma_j^{-1} (\mathbf{x} - \mathbf{m}_j) , \tag{2.12}$$

so

$$\nabla D_M^2(\mathbf{x}, \mathbf{m}_j) = 2 \Sigma_j^{-1} (\mathbf{x} - \mathbf{m}_j) , \tag{2.13}$$

and introducing (2.13) in (2.10) and solving for $\mathbf{t}$, the following is obtained

ANALYTICAL EXPRESSION

$$\mathbf{t}(\alpha) = \left(\Sigma_1^{-1} - \alpha \Sigma_2^{-1}\right)^{-1} \left(\Sigma_1^{-1} \mathbf{m}_1 - \alpha \Sigma_2^{-1} \mathbf{m}_2\right) , \tag{2.14}$$

DTC tangent point

which represents the general expression of a tangent point. On the other hand, in order to obtain the parameter expressions, (2.9) is substituted in (2.3a) and it is particularized for $\mathbf{x} = \mathbf{t}$,

$$\mathbf{w}_j = -p_j(\mathbf{t}) \cdot D_M(\mathbf{t}, \mathbf{m}_j) \cdot \nabla D_M(\mathbf{t}, \mathbf{m}_j) . \tag{2.15}$$

introducing (2.12), results in

$$\mathbf{w}_j = -p_j(\mathbf{t}) \cdot \Sigma_j^{-1}(\mathbf{t} - \mathbf{m}_j) \ . \tag{2.16}$$

Notice that the resulting discriminant is equivalent[1] if the gradient constant $p_j(\mathbf{t})$ is removed from both parameters $\mathbf{w}$ y ω_0.

[1] Two linear discriminants are equivalent if their weight vectors and biases are equally proportional.

Now, the decision rule for a linear discriminant based on a DTC can be written as follows

$$h(\mathbf{x}) = (\mathbf{w}^{\text{DTC}})^T\mathbf{x} + w_0^{\text{DTC}} \underset{C_2}{\overset{C_1}{\gtrless}} 0 \ , \tag{2.17}$$

DTC decision rule

where

$$\mathbf{w}^{\text{DTC}} = \Sigma_j^{-1}(\mathbf{t}(\alpha) - \mathbf{m}_j) \ , \tag{2.18a}$$

DTC parameters

$$\omega_0^{\text{DTC}} = -(\mathbf{w}^{\text{DTC}})^T\mathbf{t}(\alpha) \ , \tag{2.18b}$$

with j=1, 2, and $\mathbf{t}(\alpha)$ given by (2.14) with $\alpha < 0$. Note that (2.18) are the particular expressions of (2.3) when data are normally distributed or the classes are described by their first two moments. It can also be observed that a linear discriminant described by a DTC is completely determined by the tangent point $\mathbf{t}$ which, in his turn, is a function of the parameter α.

An explicit expression of $\mathbf{w}^{\text{DTC}}$ in terms of α can be obtained introducing (2.13) into (2.10), with $\mathbf{x} = \mathbf{t}$, $\mathbf{w}_1 = \alpha\,\mathbf{w}_2$, and removing $\mathbf{t}$, obtaining

$$\mathbf{w}^{\text{DTC}} = \left[\Sigma_1 - \alpha^{-1}\Sigma_2\right]^{-1}(\mathbf{m}_2 - \mathbf{m}_1) \ . \tag{2.19}$$

Simplified DTC parameters

This result allows to isolate $\mathbf{t}(\alpha)$ from (2.18a)

$$\mathbf{t}(\alpha) = \Sigma_1\left[\Sigma_1 - \alpha^{-1}\Sigma_2\right]^{-1}(\mathbf{m}_2 - \mathbf{m}_1) + \mathbf{m}_1 \ , \tag{2.20}$$

with the advantage that it only requires the calculation of one inverse and not five as in (2.14). This computational reduction applies also to the calculation of $\mathbf{w}^{\text{DTC}}$ in (2.19) and ω_0^{DTC} in (2.18b). The calculation of the DTC discriminant parameters, given α, is summarized next

Calculation of the DTC discriminant parameters:

1. Calculate the tangent point $\mathbf{t}$ (2.20)
2. Calculate the weight vector $\mathbf{w}^{\text{DTC}}$ (2.19)
3. Calculate the bias value ω_0^{DTC} (2.18b)

2.2 *Analytical relation with the parametric design*

Once the tangent point $\mathbf{t}$ is expressed as a function of α, the expression of $\mathbf{w}^{\text{DTC}}$ (2.19) is identical to that obtained by the parametric method, $\mathbf{w}^{\text{P}}$ in (1.5), if

$$s = \frac{\alpha}{\alpha - 1} \tag{2.21}$$

and

$$\mathbf{w}^{\mathrm{DTC}} = \left(\frac{\alpha}{\alpha-1}\right)\mathbf{w}^{\mathrm{P}} \,. \tag{2.22}$$

DTC METHOD
$\mathbf{w}^{\mathrm{DTC}}(\alpha)$ (2.19), $\omega_0^{\mathrm{DTC}}(\alpha)$ (2.18b)

PARAMETRIC METHOD
$\mathbf{w}^{P}(s)$ (1.5), $\omega_0^{P}(s)$ (1.7)

That is, in order to transform the $\mathbf{w}^{\mathrm{P}}$ of the parametric method into the $\mathbf{w}^{\mathrm{DTC}}$ of the DTC method or *vice versa*, first, it is necessary to change the variables s and α following (2.21), and second, multiply or divide by $(\frac{\alpha}{\alpha-1})$ respectively, as (2.22) shows.

Proof of (2.22) and (2.21).

Multiplying (2.19) by $(\frac{\alpha-1}{\alpha})$ gives

$$\left(\frac{\alpha-1}{\alpha}\right)\mathbf{w}^{\mathrm{DTC}} = \left[\frac{\alpha}{\alpha-1}\Sigma_1 - \frac{1}{\alpha-1}\Sigma_2\right]^{-1}(\mathbf{m}_2-\mathbf{m}_1) \,. \tag{2.23}$$

Defining

$$s = \frac{\alpha}{\alpha-1} \,, \tag{2.24}$$

then (2.23) becomes

$$\left(\frac{\alpha-1}{\alpha}\right)\mathbf{w}^{\mathrm{DTC}} = [s\Sigma_1 + (1-s)\Sigma_2]^{-1}(\mathbf{m}_2-\mathbf{m}_1) \,, \tag{2.25}$$

where the right side is equal to $\mathbf{w}^{\mathrm{P}}$ in (1.5) corresponding to the parametric design of linear discriminants. This proves (2.22) with (2.21).

Note that, as commented in Chapter 1.1, the optimum $\mathbf{w}^{\mathrm{P}}$ (1.5) is expressed in terms of the parameter s and does not explicitly depend on the separability criterion f.

The analytical expression of ω_0 directly depends on the separability criterion f, so each particular selection of f will produce a particular expression for ω_0. Thus, the bias value, substituting $\mathbf{t}$ (2.20) in ω_0^{DTC}(2.18b), is

Bias value ω_0

$$\omega_0^{\mathrm{DTC}}(\alpha) = -(\mathbf{w}^{\mathrm{DTC}})^T\frac{\Sigma_1\mathbf{m}_2 - \alpha^{-1}\Sigma_2\mathbf{m}_1}{\Sigma_1 - \alpha^{-1}\Sigma_2} \,. \tag{2.26}$$

2.2.1 *Bayes discriminant for Gaussian distributions*

Besides the equivalence (2.22) and (2.21) for the optimum $\mathbf{w}$, the bias value ω_0^{DTC} (2.18b), with the equivalence of s (2.21), is the same to that obtained for the Bayes linear discriminant by the parametric method ω_0^{PB} (1.11) if

PARAMETRIC METHOD
BAYES LINEAR DISC.
f: Bayes error (1.8)
SOLUTION:
$\mathbf{w}^{PB}(s^*)$ (1.5), $\omega_0^{PB}(s^*)$ (1.11)
s^*: optimum (minimum Bayes error)

$$\omega_0^{\mathrm{DTC}} = \left(\frac{\alpha}{\alpha-1}\right)\omega_0^{\mathrm{PB}} \,. \tag{2.27}$$

Therefore, the DTC discriminant is equivalent to the Bayes linear discriminant for Gaussian distributions or, according to the central limit theorem, distributions with a high dimension n, if the transformation of the parameters $\mathbf{w}^{\mathrm{DTC}}$ (2.22) and ω_0^{DTC} (2.27) is performed

from the Bayes linear discriminant parameters. Note that both parameters are multiplied by the same scalar and hence, the discriminants are equivalent.

Proof of (2.27).

Introducing (2.22) into (2.18b) and considering that $s=\frac{\alpha}{\alpha-1}$, the following is obtained

$$\omega_0^{\mathrm{DTC}} = -s(\mathbf{w}^{\mathrm{P}})^T\mathbf{t}\ . \tag{2.28}$$

In a similar way, introducing (2.22) into (2.18a) with $j=1$ and solving for $\mathbf{t}$, results in

$$\mathbf{t} = s\,\Sigma_1\mathbf{w}^{\mathrm{P}} + \mathbf{m}_1\ , \tag{2.29}$$

which, introduced into (2.28), produces

$$\omega_0^{\mathrm{DTC}} = -s^2(\mathbf{w}^{\mathrm{P}})^T\Sigma_1\mathbf{w}^{\mathrm{P}} - s(\mathbf{w}^{\mathrm{P}})^T\mathbf{m}_1\ . \tag{2.30}$$

The aim is to prove that

$$\omega_0^{\mathrm{P}} = \frac{1}{s}\,\omega_0^{\mathrm{DTC}}\ . \tag{2.31}$$

Introducing (1.11) and (2.30) into (2.31), the following is obtained

$$-\frac{s\sigma_1^2(\mathbf{w}^{\mathrm{P}})^T\mathbf{m}_2 + (1-s)\sigma_2^2(\mathbf{w}^{\mathrm{P}})^T\mathbf{m}_1}{s\sigma_1^2 + (1-s)\sigma_2^2} = -s(\mathbf{w}^{\mathrm{P}})^T\Sigma_1\mathbf{w}^{\mathrm{P}} - (\mathbf{w}^{\mathrm{P}})^T\mathbf{m}_1\ . \tag{2.32}$$

Taking into account that $\sigma_i^2 = (\mathbf{w}^{\mathrm{P}})^T\Sigma_i\mathbf{w}^{\mathrm{P}}$ (see (1.4b)), it can be easily seen that (2.32) becomes

$$(\mathbf{w}^{\mathrm{P}})^T\{[s\Sigma_1 + (1-s)\Sigma_2]\,\mathbf{w}^{\mathrm{P}} - (\mathbf{m}_2 - \mathbf{m}_1)\} = 0\ . \tag{2.33}$$

This equation is always true because the term $\{.\}$ is zero (see (1.5)). Therefore, (2.31) is true, and this proves (2.27).

2.2.2 *Fisher discriminant*

As shown in Section 1.1.2, the Fisher linear discriminant is obtained from $\mathbf{w}^{\mathrm{P}}$ (1.5) of the parametric method with $s=0.5$, which results in $\mathbf{w}^{\mathrm{PF}}$ (1.14). According to the variable change (2.21), the equivalence with DTC is $\alpha=-1$, which allows to obtain $\mathbf{w}^{\mathrm{DTC}}$ from (2.19).

$$\mathbf{w}^{\mathrm{DTC}} = \left(\frac{\alpha}{\alpha-1}\right)\mathbf{w}^{\mathrm{PF}}\ . \tag{2.34}$$

Moreover, in contrast to the classical Fisher discriminant, Fisher-DTC provides a bias parameter given by (2.18b). Substituting $\mathbf{t}$ (2.14) in ω_0^{DTC} (2.18b), or using ω_0^{DTC} (2.26), and particularizing for $\alpha=-1$, the following expression is obtained

$$\omega_0^{\mathrm{DTC}} = -(\mathbf{w}^{\mathrm{DTC}})^T\frac{\Sigma_2\mathbf{m}_1 + \Sigma_1\mathbf{m}_2}{\Sigma_2 + \Sigma_1}\ . \tag{2.35}$$

In this sense, the Fisher-DTC is a complete linear discriminant and not just an optimum projection direction.

2.2.3 *Scatter-based discriminant*

As shown in Section 1.1.3, the Scatter-based linear discriminant is obtained from $\mathbf{w}^{\mathrm{P}}$ (1.5) with $s = P_1$, which results in $\mathbf{w}^{\mathrm{PS}}$ (1.17). According to the variable change (2.21), the equivalence with DTC is $\alpha = -P_1/P_2$, *i.e.*, the ratio between the prior probabilities, allowing to obtain the direction of this discriminant $\mathbf{w}^{\mathrm{DTC}}$ (2.19) from

$$\mathbf{w}^{\mathrm{DTC}} = \left(\frac{\alpha}{\alpha - 1}\right) \mathbf{w}^{\mathrm{PS}} \,. \tag{2.36}$$

On the other hand, the DTC method provides a bias parameter by means of ω_0^{DTC}(2.18b) with $\alpha = -P_1/P_2$. Substituting $\mathbf{t}$ (2.14) in ω_0^{DTC} (2.18b), or directly using (2.26), and particularizing for that value of α, the following is obtained

$$\omega_0^{\mathrm{DTC}} = -(\mathbf{w}^{\mathrm{DTC}})^T \frac{P_1\Sigma_1\mathbf{m}_2 + P_2\Sigma_2\mathbf{m}_1}{P_1\Sigma_1 + P_2\Sigma_2} \,, \tag{2.37}$$

which is different from that obtained from the parametric method ω_0^{PS} (1.19).

2.3 *Analytical relation with the minimax convex-optimization design*

Direct method

The direct method of DTC for the calculation of the tangent point is a computational-cost competitive alternative to the iterative solution of the optimization method (SOCP). It is based on a particular result of Clark (1995), where multidimensional parameter estimators are studied that produce normal estimates. It provides (Theorem 2 of Clark, 1995) sufficient conditions for two ellipsoids to be tangent with disjoint interiors, and a procedure for finding the corresponding tangent point, presented in Section 2.5.1. This solution only requires finding the roots of a $2n$-degree polynomial.

The probability of correct classification of the points of a class, using the Mahalanobis distance $D_M(\mathbf{t}^*, \mathbf{m}_j)$ from the mean to the tangent point $\mathbf{t}^*$, is given by the cumulative distribution function

$$\varepsilon_j = \int\limits_{\mathbf{x}\in\mathcal{R}_j} \frac{1}{\sqrt{(2\pi)^n |\Sigma_j|}} \exp\left(-\frac{1}{2} D_M^2(\mathbf{x}, \mathbf{m}_j)\right) d\mathbf{x} \,, \tag{2.38}$$

where $\mathcal{R}_j = \{\mathbf{x} \in \mathbb{R}^n \mid D_M^2(\mathbf{x}, \mathbf{m}_j) \leq D_M^2(\mathbf{t}, \mathbf{m}_j)\}$ is the region that contains a probability of success ε_j for the samples of the class C_j and is defined by the set of points which have less Mahalanobis distance to the mean than the Mahalanobis distance from the mean to the tangent point $\mathbf{t}$. The probability of success as a function of the Mahalanobis distance from the mean to the tangent point is used to establish the correspondence of the DTC discriminants with the minimax convex optimization method of Section 1.2, as it is shown next.

2.3.1 *Bayes discriminant for Gaussian distributions*

The linear Bayes discriminant is obtained minimizing the Bayes error (1.8) in the projected space by the discriminant direction vector. From the Bayes' rule

$$P_1 \frac{1}{\sqrt{(2\pi)^n|\Sigma_1|}} \exp\left(-\frac{1}{2}D_M^2(\mathbf{x},\mathbf{m}_1)\right) = P_2 \frac{1}{\sqrt{(2\pi)^n|\Sigma_2|}} \exp\left(-\frac{1}{2}D_M^2(\mathbf{x},\mathbf{m}_2)\right), \tag{2.39}$$

being P_j the prior probability of the class C_j, $j=1,2$, and D_M the Mahalanobis distance given by (2.5), the optimization comes from its integration in both sides to obtain an expression dependent on the cumulative distribution function, which provides the probability of success of each class ε

$$\underbrace{P_1}_{\lambda_1}\underbrace{\int\limits_{\mathbf{x}\in\mathcal{R}_1} \frac{1}{\sqrt{(2\pi)^n|\Sigma_1|}} \exp\left(-\frac{1}{2}D_M^2(\mathbf{x},\mathbf{m}_1)\right)d\mathbf{x}}_{\varepsilon_1} + \underbrace{P_2}_{\lambda_2}\underbrace{\int\limits_{\mathbf{x}\in\mathcal{R}_2} \frac{1}{\sqrt{(2\pi)^n|\Sigma_2|}} \exp\left(-\frac{1}{2}D_M^2(\mathbf{x},\mathbf{m}_2)\right)d\mathbf{x}}_{\varepsilon_2}, \tag{2.40}$$

obtaining an expression whose maximization is equivalent to that of the optimization problem (1.21).

2.3.2 *MPDH-DTC discriminant*

The MPDH solution aims to obtain the hyperplane which separates the classes with equal and maximum probability of success (Lanckriet et al., 2002). Geometrically, the tangent point of the border of the discriminant with both ellipsoids of equal probability level is in the equidistance of Mahalanobis from the means, the point that minimizes its maximum distance to the mean of each class; *i.e.*, given an arbitrary point, the distances are calculated to each of the means of the classes and the greater of them is minimized (if during the process the greater distance is the distance to the other mean, this distance is minimized).

The relation between Mahalanobis distances from the tangent point $\mathbf{t}(\alpha)$ to each mean $\mathbf{m}_j$, $j=1,2$, satisfies

$$\frac{D_M(\mathbf{t},\mathbf{m}_1)}{D_M(\mathbf{t},\mathbf{m}_2)} = \frac{\sqrt{(\mathbf{t}-\mathbf{m}_1)^T\Sigma_1^{-1}(\mathbf{t}-\mathbf{m}_1)}}{\sqrt{(\mathbf{t}-\mathbf{m}_2)^T\Sigma_2^{-1}(\mathbf{t}-\mathbf{m}_2)}} = \frac{r_1}{r_2}. \tag{2.41}$$

If $r_1=r_2$, the DTC described by (2.41) determines a tangent point $\mathbf{t}^*$ with equal and maximum Mahalanobis distance to $\mathbf{m}_1$ and $\mathbf{m}_2$, DTC

$$D_M(\mathbf{t}^*,\mathbf{m}_1) - D_M(\mathbf{t}^*,\mathbf{m}_2) = 0. \tag{2.42}$$

Therefore, MPDH is given here by a particular DTC called MPDH-DTC discriminant, which is a non-iterative alternative to optimization methods used in MPM.

As stated before, the Mahalanobis equidistance from the tangent point to both means implies that the discriminant separates both classes with equal probability. Thus, the probability of success enclosed by a given Mahalanobis distance $D_M(\mathbf{t}^*, \mathbf{m}_j)$ is given by the cumulative distribution function (2.38). Considering $D_M(\mathbf{t}^*, \mathbf{m}_1) = D_M(\mathbf{t}^*, \mathbf{m}_2)$, the probability of success of each class is equal too, verifying

Minimax optimization method

$$\underbrace{\int\limits_{\mathbf{x}\in\mathcal{R}_1} \frac{1}{\sqrt{(2\pi)^n|\Sigma_1|}} \exp\left(-\frac{1}{2}D_M^2(\mathbf{x},\mathbf{m}_1)\right)d\mathbf{x}}_{\varepsilon_1} = \underbrace{\int\limits_{\mathbf{x}\in\mathcal{R}_2} \frac{1}{\sqrt{(2\pi)^n|\Sigma_2|}} \exp\left(-\frac{1}{2}D_M^2(\mathbf{x},\mathbf{m}_2)\right)d\mathbf{x}}_{\varepsilon_2} . \tag{2.43}$$

Thus, if $\varepsilon_1 = \varepsilon_2 = \varepsilon$ and it is aimed to maximize the success probability ε, there is an equivalence with the expression of the optimization problem (1.22).

2.4 *Quasi-Bayes-DTC discriminant*

In this section a new discriminant DTC is presented which, due to its design characteristics, has been called Quasi-Bayes-DTC. The DTC tangent point that defines the discriminant is the crossing point between the optimal Bayes border for Gaussian distributions and the curve of all DTC points, Figure 2.4. As will be seen in the experiments section, it produces accuracy results very close to those of the Bayes linear discriminant for Gaussian distributions with a lower computational cost because it is a direct solution (no search for any parameter is required). Finally, the relationship of this discriminant with the discriminators obtained through minimax convex-optimization will be demonstrated.

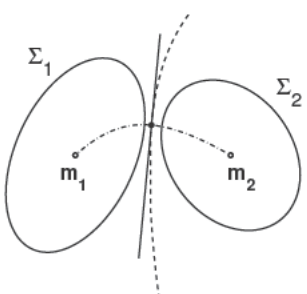


Figure 2.4: Quasi-Bayes DTC linear discriminant. This discriminant (bold curve) is described by the DTC tangent point given by the intersection between the Bayes quadratic border (dashed line) and the DTC curve of all possible tangent points (dotted and dashed curve). The Quasi-Bayes DTC tangent point is shown as a bold dot.

The calculation of the discriminant is similarly done to the MPDH-DTC, but the equidistance of Mahalanobis is replaced by a not-null difference of Mahalanobis distances. The Quasi-Bayes-DTC tangent point $\mathbf{t}^*$, because it is also the tangent point of the quadratic Bayes border, satisfies Bayes' rule (2.39).

Similarly to (2.42), taking logarithms in (2.39)

$$D_M^2(\mathbf{t}^*, \mathbf{m}_1) - D_M^2(\mathbf{t}^*, \mathbf{m}_2) = K\,, \tag{2.44}$$

where

$$K = \ln\left[\left(\frac{P_1}{P_2}\right)^2 \frac{|\Sigma_2|}{|\Sigma_1|}\right] . \tag{2.45}$$

This discriminant provides a good classification in terms of error probability when the prior probabilities are an important part of the problem –as seen in imbalanced problems with extreme a priori probability values– since it always moves with the optimal non-linear Bayes discriminant.

In order to establish the relation with the minimax optimization

Minimax optimization method

method, $D'^2_M(\mathbf{t}^*, \mathbf{m}_2) = D^2_M(\mathbf{t}^*, \mathbf{m}_2) + K$ being a new Mahalanobis distance from the mean of the class C_2 which is equal to the Mahalanobis distance from the mean of the class C_1 to the tangent point of the discriminant

$$D^2_M(\mathbf{t}^*, \mathbf{m}_1) = D'^2_M(\mathbf{t}^*, \mathbf{m}_2) \; . \tag{2.46}$$

The success probability bounded by these Mahalanobis distances is

$$\underbrace{\int\limits_{\mathbf{x}\in\mathcal{R}_1} \frac{1}{\sqrt{(2\pi)^n|\Sigma_1|}} \exp\left(-\frac{1}{2}D^2_M(\mathbf{x},\mathbf{m}_1)\right)d\mathbf{x}}_{\varepsilon_1} = \underbrace{\int\limits_{\mathbf{x}\in\mathcal{R}_2} \frac{1}{\sqrt{(2\pi)^n|\Sigma_2|}} \exp\left(-\frac{1}{2}D'^2_M(\mathbf{x},\mathbf{m}_2)\right)d\mathbf{x}}_{\varepsilon'_2} \; , \tag{2.47}$$

where

$$\varepsilon'_2 = \int\limits_{\mathbf{x}\in\mathcal{R}_2} \frac{1}{\sqrt{(2\pi)^n|\Sigma_2|}} \exp\left(-\frac{1}{2}\left(D^2_M(\mathbf{x},\mathbf{m}_2)+K\right)\right)d\mathbf{x} = \Gamma\cdot\varepsilon_2, \tag{2.48}$$

being ε_2 described by (2.38) and resulting in $\varepsilon_1 = \Gamma\cdot\varepsilon_2$, with

$$\Gamma = \frac{P_2}{P_1}\sqrt{\frac{|\Sigma_1|}{|\Sigma_2|}} \; . \tag{2.49}$$

Thus, the optimization problem, following its general expression (1.20), and using the results obtained in MPDH (2.43)-(1.22) is

$$\begin{aligned} \max_{\varepsilon,\mathbf{w}\neq\mathbf{0},\omega_0} \; \varepsilon \quad s.t. \quad & \inf_{\mathbf{x}\sim(\mathbf{m}_1,\Sigma_1)} P\{\mathbf{w}^T\mathbf{x}+\omega_0 \geq 0\} \geq \varepsilon \\ & \inf_{\mathbf{x}\sim(\mathbf{m}_2,\Sigma_2)} P\{\mathbf{w}^T\mathbf{x}+\omega_0 \leq 0\} \geq \Gamma\cdot\varepsilon \; , \end{aligned} \tag{2.50}$$

QUASI-BAYES (minimax optimization method)

2.5 *Calculation of the DTC discriminants*

The procedure of a DTC calculation is summarized next. First, the first two moments of the class distributions are estimated. Second, the parameter α is calculated, which determines an unique DTC discriminant. This calculation can come from the calculation of the s parameter by means of the parametric method and then establish the correspondence with the α of the DTC method; or, in the case of MPDH and Quasi-Bayes, the α value is calculated through the Clark polynomial, as shown in Section 2.5.1. Once α is known, is direct the calculation of the tangent point $\mathbf{t}$ is direct and then, the parameters $\mathbf{w}^{\text{DTC}}$ and ω_0^{DTC}, which determine the DTC discriminant. Section 2.5.2 presents the DTC algorithm more exhaustively.

Calculation of the DTC discriminants:

1. Estimation of $\mathbf{m}_1$, $\mathbf{m}_2$, Σ_1 and Σ_2 from data.
2. Determine α^*:
   - From the parametric method
     - Applying the transformation $s \to \alpha$ (2.22)
   - Through the Clark polynomial
     - MPDH problem with $K=0$
     - Quasi-Bayes discriminant with K of (2.45)
3. Calculate the tangent point $\mathbf{t}(\alpha^*)$ (2.20)
4. Calculate the weight vector $\mathbf{w}^{\text{DTC}}$ (2.19) or (2.18a)
5. Calculate the bias value ω_0^{DTC} (2.18b)

2.5.1 *Calculation of α by the Clark polynomial*

Let's $\mathbf{m}_j \in \mathbb{R}^n$ the means vectors and $\Sigma_j \in \mathbb{R}^{n\times n}$, j=1, 2, the covariance matrices of a given binary classification problem. The aim is to find the DTC tangent point. First, data are spatially moved in such a way that one of the means, *e.g.*, $\mathbf{m}_2$, is $\mathbf{0}$. This is done by subtracting $\mathbf{m}_2$ to all data points. Then, the matrix $T \in \mathbb{R}^{n\times n}$ simultaneously diagonalize Σ_1^{-1} and Σ_2^{-1} to transform them into a positive definite diagonal matrix, $D \in \mathbb{R}^{n\times n}$, and the identity matrix $I \in \mathbb{R}^{n\times n}$

$$T^T \Sigma_1^{-1} T = I \,, \tag{2.51a}$$
$$T^T \Sigma_2^{-1} T = D \,. \tag{2.51b}$$

Now, each data point $\mathbf{x}$ has been transformed into $\tilde{\mathbf{x}}$ according to

$$\tilde{\mathbf{x}} = T^{-1}(\mathbf{x} - \mathbf{m}_2) \,, \tag{2.52}$$

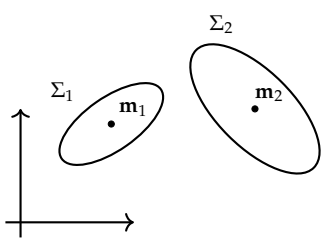
and the transformed means are

$$\tilde{\mathbf{m}}_1 = T^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \,, \tag{2.53a}$$
$$\tilde{\mathbf{m}}_2 = T^{-1}(\mathbf{m}_2 - \mathbf{m}_2) = 0 \,. \tag{2.53b}$$

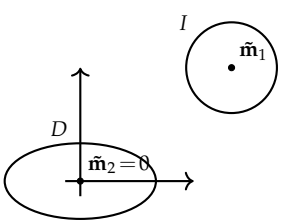
So, the new problem is finding the tangent point between a hypersphere centered at $\tilde{\mathbf{m}}_1$ and an ellipsoid centered at the origin $\tilde{\mathbf{m}}_2 = 0$

Theorem 2 of Clark (1995) gives sufficient conditions for two ellipsoids to be tangent with disjoint interiors and a procedure to find the corresponding tangent point. These sufficient conditions are

$$\tilde{\mathbf{t}}(\alpha) = (I - \alpha D)^{-1} \tilde{\mathbf{m}}_1 \,, \tag{2.54a}$$
$$r_1^2 = (\tilde{\mathbf{t}}(\alpha) - \tilde{\mathbf{m}}_1)^T (\tilde{\mathbf{t}}(\alpha) - \tilde{\mathbf{m}}_1) \,, \tag{2.54b}$$
$$r_2^2 = \tilde{\mathbf{t}}(\alpha)^T D\, \tilde{\mathbf{t}}(\alpha) \,, \tag{2.54c}$$
$$\alpha < 0 \,, \tag{2.54d}$$



(a) Original space



(b) Transformed space

Figure 2.5: Simultaneous diagonalization

where $\tilde{\mathbf{t}}$ is the DTC tangent point in the transformed space. The unique value α^* that satisfies the conditions is the real and negative solution of

$$G(\alpha) - H(\alpha) = K \,, \tag{2.55}$$

where

$$K \;= \tilde{D}_M^2(\tilde{\mathbf{t}}, \tilde{\mathbf{m}}_1) - \tilde{D}_M^2(\tilde{\mathbf{t}}, \tilde{\mathbf{m}}_2) \tag{2.56}$$

is the difference of the Mahalanobis distances from the tangent point to the means and

$$G(\alpha) = \alpha^2 \, \tilde{\mathbf{m}}_1^T \, D^2 \, (I - \alpha D)^{-2} \, \tilde{\mathbf{m}}_1 \,, \tag{2.57a}$$

$$H(\alpha) = \tilde{\mathbf{m}}_1^T \, D \, (I - \alpha D)^{-2} \, \tilde{\mathbf{m}}_1 \,. \tag{2.57b}$$

To find the solution, $G(\alpha)$ and $H(\alpha)$ are expressed as

$$G(\alpha) = \frac{\alpha^2 \, C(\alpha)}{A(\alpha)} \,, \qquad H(\alpha) = \frac{B(\alpha)}{A(\alpha)} \tag{2.58}$$

where $A(\alpha)$ is a $2n$-degree polynomial, and $B(\alpha)$ and $C(\alpha)$ are polynomials of degree $2n-2$ or less. Using this result, the solution of (2.55) is the unique real root of the $2n$-degree Clark polynomial

$$\alpha^2 \, C(\alpha) - B(\alpha) - K A(\alpha) = 0 \,, \tag{2.59}$$

which satisfies $\alpha < 0$. $A(\alpha)$, $B(\alpha)$ and $C(\alpha)$ can be computationally obtained using Algorithm 1 of Clark (1995).

Once the solution α^* is obtained, the DTC tangent point $\tilde{\mathbf{t}}(\alpha^*)$ in the transformed space is calculated by (2.54a), and the corresponding tangent point $\mathbf{t}$ in the original space can be obtained from (2.52) with $\mathbf{x} = \mathbf{t}(\alpha^*)$, $\tilde{\mathbf{x}} = \tilde{\mathbf{t}}(\alpha^*)$, and isolating $\mathbf{t}$

$$\mathbf{t}(\alpha^*) = T \, \tilde{\mathbf{t}}(\alpha^*) + \mathbf{m}_2 \,. \tag{2.60}$$

This result can also be obtained from (2.14) or (2.20) with α^*. This is true because (2.54) can be obtained from the DTC analysis.

Proof. Clark conditions (2.54) can be obtained from the DTC analysis in the transformed space.

Considering the Mahalanobis distances in the transformed space from the tangent point $\tilde{\mathbf{t}}$ to the means $\tilde{\mathbf{m}}_j$, j=1, 2, ($\tilde{\mathbf{m}}_2 = 0$),

$$\tilde{D}_M(\tilde{\mathbf{t}}, \tilde{\mathbf{m}}_j) = r_j \,, \tag{2.61}$$

note that the conditions (2.54b) and (2.54c) are the following Mahalanobis distances particularized in $\tilde{\mathbf{x}} = \tilde{\mathbf{t}}$

$$\tilde{D}_M^2(\tilde{\mathbf{x}}, \tilde{\mathbf{m}}_1) = (\tilde{\mathbf{x}} - \tilde{\mathbf{m}}_1)^T (\tilde{\mathbf{x}} - \tilde{\mathbf{m}}_1) \,, \tag{2.62a}$$

$$\tilde{D}_M^2(\tilde{\mathbf{x}}, \mathbf{0}) = \tilde{\mathbf{x}}^T D \, \tilde{\mathbf{x}} \,, \tag{2.62b}$$

and applying the gradient operator and particularizing in $\tilde{\mathbf{x}} = \tilde{\mathbf{t}}$

$$\nabla \tilde{D}_M^2(\tilde{\mathbf{t}}, \tilde{\mathbf{m}}_1) = 2(\tilde{\mathbf{t}} - \tilde{\mathbf{m}}_1) \,, \tag{2.63a}$$

$$\nabla \tilde{D}_M^2(\tilde{\mathbf{t}}, \mathbf{0}) = 2 \, D \, \tilde{\mathbf{t}} \,. \tag{2.63b}$$

Using (2.10) from the DTC analysis

$$\nabla \tilde{D}_M^2(\tilde{\mathbf{t}}(\alpha), \tilde{\mathbf{m}}_1) = \alpha \cdot \nabla \tilde{D}_M^2(\tilde{\mathbf{t}}(\alpha), \mathbf{0}) \; . \tag{2.64}$$

Substituting (2.63) in (2.64)

$$2(\tilde{\mathbf{t}} - \tilde{\mathbf{m}}_1) = 2\,\alpha D\,\tilde{\mathbf{t}} \; , \tag{2.65}$$

and isolating $\tilde{\mathbf{t}}$

$$\tilde{\mathbf{t}} = (I - \alpha D)^{-1}\,\tilde{\mathbf{m}}_1 \; , \tag{2.66}$$

which is the same as condition (2.54a), proving the Clark conditions can be obtained from the DTC analysis.

Note that the solution α^* of the Clark polynomial (2.59) is the solution of the MPDH problem if $K = 0$ (2.56), because the difference of Mahalanobis distances in the tangent point is null, $r_1 = r_2 = r$. On the other hand, the solution of the Clark polynomial is the Quasi-Bayes discriminant if the difference of Mahalanobis distances K in the tangent point satisfies the Bayes' rule, calculated in (2.45).

2.5.2 *DTC algorithm*

This method to determine the DTC discriminant follows the calculation of the tangent point of Section 2.5.1 by the Clark polynomial.

The algorithm's input is the labeled patterns and the output are the DTC discriminant parameters $\mathbf{w}^*$ and ω_0^*.

First, the first two moments of the classes and the prior probabilities are estimated from data: $\mathbf{m}_j$, Σ_j, and P_j, $j = 1, 2$. Then, the transformation matrix T (Section 2.5.1) is calculated using the Cholesky factorization and the Schur decomposition in order to simultaneously diagonalize both inverse covariance matrices. The result of the Cholesky factorization of the covariance matrix Σ_1 is the matrix G, which satisfies

$$GG^T = \Sigma_1^{-1} \; . \tag{2.67}$$

Using

$$F = G^{-1}\Sigma_2^{-1}(G^T)^{-1} \; , \tag{2.68}$$

the Schur decomposition of F results in Q and Z, Z being a diagonal matrix, which satisfy

$$Q^T F Q = Z \; . \tag{2.69}$$

Then, it is possible to calculate the transformation matrix T as

$$T = (G^T)^{-1} Q \; . \tag{2.70}$$

The diagonal matrix and the identity matrix after the simultaneous diagonalization are

$$D = T^T \Sigma_2^{-1} T \; , \tag{2.71a}$$

$$I = T^T \Sigma_1^{-1} T \; , \tag{2.71b}$$

D being a positive definite diagonal matrix. The Algorithm 1 of Clark (1995) allows to construct the polynomial in α to obtain the DTC tangent point. First, the auxiliary vectors b, the diagonal of the diagonal matrix D, and y, the not null mean in the transformed space, are calculated,

$$\mathbf{b} = \text{diag}(D) \,, \tag{2.72a}$$
$$\mathbf{y} = T^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \,. \tag{2.72b}$$

Hereinafter, the subscripts numbers refer to the positions of a vector, except $\mathbf{x}_j$ and Σ_j where the subscripts still refer to the classes, j=1, 2. b and y are used to calculate the coefficients of the polynomial in α

$$C_1(\alpha) = \mathbf{b}_1^2 \mathbf{y}_1^2 \,, \tag{2.73a}$$
$$B_1(\alpha) = \mathbf{b}_1 \mathbf{y}_1^2 \,, \tag{2.73b}$$
$$A_1(\alpha) = \mathbf{b}_1^2 \alpha^2 - 2\mathbf{b}_1 \alpha + 1 \,, \tag{2.73c}$$

to iteratively calculate, from k=2 to n, being n the dimension of data,

$$\begin{aligned} C_k(\alpha) &= \mathbf{b}_k^2 \alpha^2 C_{k-1}(\alpha) - 2\mathbf{b}_k \alpha C_{k-1}(\alpha) \\ &\quad + C_{k-1}(\alpha) + \mathbf{b}_k^2 \mathbf{y}_k^2 A_{k-1}(\alpha) \,, \end{aligned} \tag{2.74a}$$
$$\begin{aligned} B_k(\alpha) &= \mathbf{b}_k^2 \alpha^2 B_{k-1}(\alpha) - 2\mathbf{b}_k \alpha B_{k-1}(\alpha) \\ &\quad + B_{k-1}(\alpha) + \mathbf{b}_k \mathbf{y}_k^2 A_{k-1}(\alpha) \,, \end{aligned} \tag{2.74b}$$
$$\begin{aligned} A_k(\alpha) &= \mathbf{b}_k^2 \alpha^2 A_{k-1}(\alpha) - 2\mathbf{b}_k \alpha A_{k-1}(\alpha) \\ &\quad + A_{k-1}(\alpha) \,. \end{aligned} \tag{2.74c}$$

Then, solving the $2n$-degree Clark polynomial (2.59) in α

$$\alpha^2 C(\alpha) - B(\alpha) - KA(\alpha) = 0 \,, \tag{2.75}$$

to obtain a unique real negative α^*, which is the solution of the DTC tangent point $\mathbf{t}^*$ given by (2.14)

$$\mathbf{t}^* = \left(\Sigma_1^{-1} - \alpha^* \Sigma_2^{-1}\right)^{-1} \left(\Sigma_1^{-1} \mathbf{m}_1 - \alpha^* \Sigma_2^{-1} \mathbf{m}_2\right) \,. \tag{2.76}$$

Using the tangent point $\mathbf{t}^*$ in (2.18)

$$\mathbf{w}^* = \Sigma_1^{-1}(\mathbf{t}^* - \mathbf{m}_1) \,, \tag{2.77a}$$
$$\omega_0^* = -(\mathbf{w}^*)^T \mathbf{t}^* \,, \tag{2.77b}$$

the parameters $\mathbf{w}^*$ and ω_0^* of the DTC linear discriminant are obtained.

Note that the solution α^* of the Clark polynomial (2.59) is the solution of the MPDH problem if K = 0 (2.56). On the other hand, the solution of the Clark polynomial is the Quasi-Bayes discriminant if the difference of Mahalanobis distances K in the tangent point satisfies the Bayes' rule, being calculate in (2.45).

Next, the DTC algorithm is presented. Note that A, B, C and P are vectors and n is the data dimension.

Algorithm 2: DTC algorithm

Data: Labeled patterns
Result: $\mathbf{w}^*$, ω_0^*
Estimation of $\mathbf{m}_j$, Σ_j and P_j, j=1, 2
$G = \text{cholesky}(\Sigma_1^{-1})$
$F = G^{-1}\Sigma_2^{-1}(G^T)^{-1}$
$Q, Z = \text{schur}(F)$
$T = (G^T)^{-1}Q$
$D = T^T\Sigma_2^{-1}T$
$\mathbf{b} = \text{diag}(D)$
$\mathbf{y} = T^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$
$C = \mathbf{b}_1^2\mathbf{y}_1^2$
$B = \mathbf{b}_1\mathbf{y}_1^2$
$A = [\mathbf{b}_1^2,\ -2\mathbf{b}_1,\ 1]$
for k=2 *to* n **do**
  $C = \mathbf{b}_k^2 \cdot [C,\ 0,\ 0] - 2\mathbf{b}_k \cdot [0,\ C,\ 0] + [0,\ 0,\ C] + \mathbf{b}_k^2\mathbf{y}_k^2 \cdot A$
  $B = \mathbf{b}_k^2 \cdot [B,\ 0,\ 0] - 2\mathbf{b}_k \cdot [0,\ B,\ 0] + [0,\ 0,\ B] + \mathbf{b}_k\mathbf{y}_k^2 \cdot A$
  $A = \mathbf{b}_k^2 \cdot [A,\ 0,\ 0] - 2\mathbf{b}_k \cdot [0,\ A,\ 0] + [0,\ 0,\ A]$
if *MPDH-DTC* **then**
  $K = 0$
if *Quasi-Bayes-DTC* **then**
  $K = -2\log\left(\frac{P_1}{(1-P_1)}\sqrt{\frac{|\Sigma_2|}{|\Sigma_1|}}\right)$
$P = [C,\ 0,\ 0] - [0,\ 0,\ B] + K \cdot A$
$\alpha^* = \text{roots}(P)$
$\mathbf{t}^* = \left(\Sigma_1^{-1} - \alpha^*\Sigma_2^{-1}\right)^{-1} * \left(\Sigma_1^{-1}\mathbf{m}_1 - \alpha^*\Sigma_2^{-1}\mathbf{m}_2\right)$
$\mathbf{w}^* = \Sigma_1^{-1}(\mathbf{t}^* - \mathbf{m}_1)$
$\omega_0^* = -(\mathbf{w}^*)^T\mathbf{t}^*$

Part II

Non-linear DTC discriminants

In some cases, linear discriminants are used to solve classification problems due to their simplicity. However, linear classifiers are often insufficient to effectively solve many other problems, so that more powerful non-linear architectures are needed.

A non-linear discriminant could be constructed by transforming the input data into an **intermediate space** and then using a linear discriminant since it has been shown that, through an appropriate transformation, the new data set in the transformed space has a greater linear separability (Vapnik, 1995; Huang et al., 2006).

This transformation can be done with direct methods such as Principal Components Analysis (PCA) by Jollife (2002), paradigm of dimensionality reduction, or by more or less complex neural structures. Among them, the Single-hidden Layer Feedforward Networks (SLFNs) (Huang et al., 2006; Widrow et al., 2013; Martínez-García and Sancho-Gómez, 2018), which perform the transformation through a hidden layer with non-linear units, allowing to solve the classification problem in the output layer with a linear discriminant. The most representative SLFNs are: 1) the Multi-layer Perceptron (MLP) with a hidden layer of sigmoid units or Rectified Linear Units (ReLU). These networks are normally trained using gradient algorithms (Back-propagation) and are designed using Cross-Validation (CV) techniques. They present the drawback of training stagnation (either by falling into a local minimum, or by the effect known as paralysis) and a computationally expensive design;

2) The Radial Basis Function Networks (**RBFN**) with a hidden layer of Gaussian kernels. Vector quantization techniques are used for kernel calculation and Least Mean Squares (LMS) for the output layer weights. The design of the hidden layer size is done by CV. There are other alternative ways of designing and training an RBF network but the one described stands out for its effectiveness (Haykin, 2009); finally, 3) kernel-based networks that use the so-called kernel trick to perform the non-linear transformation in order to design and train the complete network by solving a convex optimization problem (Schölkopf and Smola, 2001). These are, among others, the Support Vector Machines (SVMs). The main drawback of this method is its high computational cost for a large number of samples.

There also are deep networks which perform a transformation of the data space to overcome the performance obtained in the original space. Highlighted among others, the Stacked Denoising Auto-Encoders (SDAE) (Vincent et al., 2010), a very often used technique due to its high representation capability; Generative Adversarial Nets (GAN) (Goodfellow et al., 2014), and more concrete Adversarial Auto-Encoders (AAE) (Makhzani et al., 2015); and Variational Auto-Encoders (VAE) (Doersch, 2016), in which the intermediate space can be used to apply a linear discriminant.

3

Radial Basis Function Networks (RBFN)

Non-linear projection

In this chapter, the original space of the input data $\mathbf{x} \in \mathbb{R}^n$ of the classification problem is mapped to a higher-dimensional feature space via the mapping function $\phi : \mathbb{R}^n \to \mathbb{R}^m$, such that the linear decision boundary $\mathcal{H}_{\text{L}}(\mathbf{w}, \omega_0) = \{\phi(\mathbf{x}) \in \mathbb{R}^m \mid \mathbf{w}^T\phi(\mathbf{x}) + \omega_0 = 0\}$ in the projected feature space $\mathbb{R}^m$ corresponds to the non-linear decision boundary in the original data space $\mathcal{H}_{\text{NL}}(\mathbf{w}, \omega_0) = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{w}^T\phi(\mathbf{x}) + \omega_0 = 0\}$.

SLFN

It is well known that an appropriate non-linear projection into an intermediate feature space increases the linear separability, allowing the use of a linear discriminant (Vapnik, 1995; Huang et al., 2006). A Single-hidden Layer Feedforward Neural Network (SLFN) (Huang et al., 2006; Widrow et al., 2013; Martínez-García and Sancho-Gómez, 2018) is a neural network architecture based on this idea to construct a non-linear discriminant, in which the input data are projected by the hidden layer and are classified by means of the output linear discriminant. Therefore, the non-linear behavior is provided by the hidden layer through the different types of kernels. This network transforms the N input samples in $\mathbb{R}^n$ into N samples in $\mathbb{R}^m$, m being the number of hidden nodes. The number of classes determines the output layer dimension, which is, in turn, the number of output nodes or linear discriminants. However, note that in binary classification problems the output is one-dimensional and hence, only one linear discriminant is needed in the output layer.

Training

From a particular linear discriminant, it is possible to build a non-linear classifier using an SLFN architecture. For this, training is required for the weights of the hidden layer (the non-linear part) and for those of the output layer (the linear part). While MLP and SVM employ training algorithms for the whole architecture –with a good performance but with the local minima stuck issue in the case of the gradient descent training (Back-propagation)– other SLFNs allow an independent training for each layer (Haykin, 2009; Duda et al., 2000)

with an unsupervised training, *i.e.*, regardless of the targets, for the hidden layer and supervised for the output layer. The most common unsupervised training of the hidden layer weights is their random fixing at the beginning, like ELM (Extreme Learning Machine) (Huang et al., 2006) and the No-Propagation algorithm (Widrow et al., 2013; Martínez-García and Sancho-Gómez, 2018), which increase the training speed but sometimes needs a larger hidden layer. Vector quantization techniques for Gaussian transformations present a more precise unsupervised training, as will be seen later. Once the hidden layer is trained, the linear discriminant of the output layer can be trained by pseudoinverse, *e.g.*, ELM; by a gradient algorithm, *e.g.*, mean squares (LMS) of No-propagation, less sensitive to noise than pseudoinverse (Martínez-García and Sancho-Gómez, 2018); or by a parametric algorithm, *e.g.*, the DTC discriminants.

Unsupervised (non-linear hidden layer)

- Random weights
- Vector quantization

Supervised (linear output layer)

- Pseudoinverse
- Gradient descent
- Parametric methods

KERNELS

In order to perform the mapping to the feature space, there are memory-based methods in which the training samples, or a subset of them, are used to predict in the test phase (Bishop, 2006). They consists of linear combinations of a kernel function which includes a similarity measurement of two samples in the input space. This method is fast in training but slow in testing. For non-linear mapping functions $\phi(\mathbf{x})$, the kernel is given by

$$\kappa(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}') \ , \tag{3.1}$$

$\mathbf{x}$ and $\mathbf{x}'$ being two samples in the input space.

A necessary and sufficient condition for a function $\kappa(\mathbf{x}, \mathbf{x}')$ to be a valid kernel is that the Gram matrix is positive definite. The Gram matrix is a symmetric $L_{[N\times N]}$ matrix defined as

$$L = \Phi \, \Phi^T \ , \tag{3.2}$$

where Φ is the design matrix whose i-th row is given by $\phi(\mathbf{x}^{(i)})^T$, $i=1, 2, \ldots, N$. Notice the kernel formulation deals with N-dimensional inverse square matrices while the initial space deals with m-dimensional inverse square matrices, where the number of samples N is typically much larger than the dimension of samples m. Therefore, the disadvantage of the kernel formulation is the increase in computational cost, but with the advantage of working directly in terms of kernels and not with the explicit formulation of the feature vector $\phi(\mathbf{x})$, which allows to use feature spaces of high, even infinite, dimensionality.

High computational cost

Any algorithm which can be expressed with an inner product in the feature space can also be expressed with a kernel, what is called the kernel trick or kernel substitution.

Kernel transformation

There are many possible kernel types, some of them are described

below with their parameters

$$\kappa(\mathbf{x},\mathbf{x}') = \mathbf{x}^T\mathbf{x}' \qquad \text{Linear}\ , \tag{3.3a}$$

$$\kappa(\mathbf{x},\mathbf{x}') = (a\mathbf{x}^T\mathbf{x}' + b)^r \qquad \text{Polynomial}\ (a,b,r)\ , \tag{3.3b}$$

$$\kappa(\mathbf{x},\mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right) \qquad \text{Gaussian}\ (\sigma)\ , \tag{3.3c}$$

$$\kappa(\mathbf{x},\mathbf{x}') = \tanh(a\mathbf{x}^T\mathbf{x}' + b) \qquad \text{Sigmoid}\ (a,b)\ . \tag{3.3d}$$

RADIAL BASIS FUNCTIONS

The kernel functions which depend on the distance, typically Euclidean, between the input samples $\kappa(\mathbf{x},\mathbf{x}') = \kappa(\|\mathbf{x}-\mathbf{x}'\|)$ are known as radial basis functions, and more specifically, Gaussian radial basis functions if the kernel is Gaussian (3.3c). Note that the normalization coefficient is omitted due to it not being a probability density. Hereinafter, the radial basis functions will be considered Gaussian.

Originally, radial basis functions were used for exact interpolation: given a set of input samples $\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$, the goal is to find a smooth function $g(\mathbf{x})$ that is a linear combination or superposition of radial basis functions centered on each sample

$$g(\mathbf{x}) = \sum_{i=1}^{N} w_i \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}^{(i)}\|^2}{2\sigma^2}\right)\ , \tag{3.4}$$

that fits every target or class value exactly. The weights w_i are fitted by least mean squares (LMS). However, target values are generally noisy and exact interpolation is undesirable because it produces an over-fitted solution. Besides, for large data sets, it is very costly to evaluate.

Not exact interpolation

A modification consists of reducing the number of basis functions to provide a smooth, not exact, interpolation function in which the number of basis functions is related to the complexity of the problem. Now, the centers of the basis functions are no longer the samples, but representative centroids of the samples; and the width of the basis functions σ becomes different and trainable for each basis function. It is also possible to add a bias parameter to compensate for the difference with the target values and include it in the summation by an extra bias function $\phi_0 = 1$. Thus, the resulting Gaussian radial basis functions are

$$\phi_k(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\boldsymbol{\mu}_k\|^2}{2\sigma_k^2}\right)\ , \tag{3.5}$$

$\boldsymbol{\mu}_k$ being the centers of the basis functions, k=1, 2, ..., m, m being the number of the Gaussian basis functions. Note that the equation is valid for any arbitrary covariance matrix.

Universal approximator

Like multilayer perceptrons, radial basis function networks are universal approximators (Bishop, 1995). Even with mild restrictions on the form of the kernels, the universal approximation property

still holds. Radial basis function networks also presents the property of being the "best approximation", consisting of the existence of a function, among all possible functions with adjustable parameters, which provides the minimum approximation error for any given function.

Fast training. Two stages

The training of radial basis function networks can be faster that the training of multilayer perceptrons because it is possible to apply a two-stage training procedure, as mentioned. In multilayer perceptrons, the trainable parameters are usually fitted at the same time in a global optimization procedure that uses supervised training. On the contrary, in radial basis functions, in the first stage, the parameters of the basis functions, *i.e.*, their position $\boldsymbol{\mu}$ and their width σ in the case of Gaussian functions, are fitted by unsupervised methods, only considering the input samples, which make them fast methods. Therefore, the basis function centers $\boldsymbol{\mu}$ can be considered as prototypes of the input samples. In the second stage, the output layer weights are fitted solving a linear problem that minimizes an error function, which is also fast. This allows the use of a large amount of unlabeled training data for the calculation of the parameters of the basis functions and a small quantity for the labeled training of the output weights, keeping the number of samples per number of parameters ratio high for each training stage.

3.1 RBF-DTC

It is a non-linear DTC discriminant formed by a Gaussian RBF network (3.3c), which takes advantage of the separate pre-design of the hidden layer, with a linear discriminant DTC in the output layer. It is necessary for an effective selection of the centroids, the parameter that places the Gaussians in the input space, to provide to the posterior linear discriminant an appropriate representation of the data distribution with a high expressive capacity. Thus, the hidden layer is pre-trained using vector quantization techniques such as Frequency Sensitive Competitive Learning (FSCL), (Ahalt et al., 1990), described in Appendix C. Finally, the linear DTC discriminant of the output layer is trained like a standalone linear discriminant. The non-linear discriminant constructed in this way preserves the properties of the linear discriminant from which it originates.

The architecture of this RBF-DTC network is shown in Figure 3.1, in which the n-dimensional input patterns, $\mathbf{x}^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}\}$ or $X_{[N\times n]}$ in a matrix form, $i{=}1, 2, \ldots, N$, N being the number of samples, are transformed into N m-dimensional hidden patterns by means of m Gaussian functions centered at its centroids. Each centroid $\mathbf{c}^{(k)} = \{c_1^{(k)}, c_2^{(k)}, \ldots, c_n^{(k)}\}$, $k{=}1, 2, \ldots, m$, denotes the position of the k-th Gaussian node in the input space, having $n{+}1$ parameters to be fixed: The n parameters of $\mathbf{c}^{(k)}$ and the standard deviation.

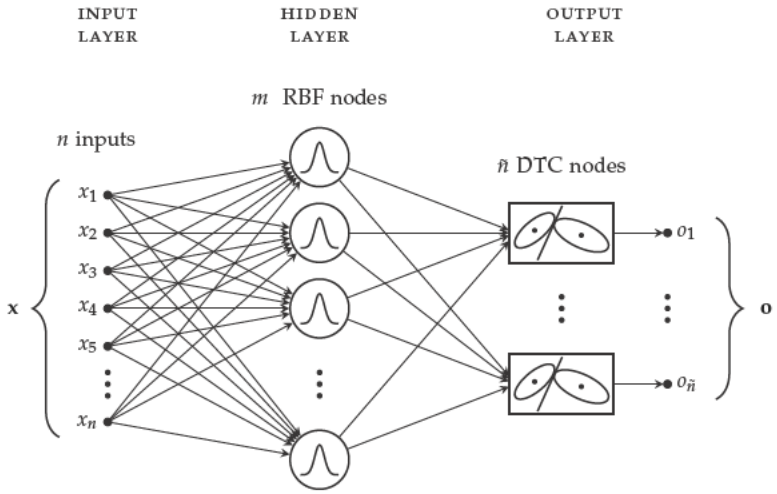


Figure 3.1: Architecture of the non-linear RBF-DTC discriminant: $\mathbf{x}$ input patterns, RBF nodes in the hidden layer, DTC nodes in the output layer and $\mathbf{o}$ output patterns.

The use of Gaussian nodes with radial symmetry provides the best option in terms of the trade-off between computational cost and classification accuracy (Haykin, 2009). Using this approach, the output of the k-th Gaussian node for the i-th input sample, $H_{[N\times m]}$ in a matrix form, is given by

$$h_{i,k} = \frac{1}{\sigma_k\sqrt{2\pi}} \exp\left(-\frac{\|\mathbf{x}^{(i)} - \mathbf{c}^{(k)}\|^2}{2\sigma_k^2}\right), \tag{3.6}$$

with $i=1,2,\ldots,N$, and $k=1,2,\ldots,m$. For a multi-class problem, the j-th component of the network output for the i-th input sample, $O_{[N\times\tilde{n}]}$ in a matrix form, is given by the linear combination of the hidden layer outputs and the weights of the DTC linear discriminant as follows, including the classification rule,

$$o_{i,j} = (\mathbf{w}^{(j)})^T\mathbf{h}^{(i)} + \omega_0{}^{(j)} \underset{C_2}{\overset{C_1}{\gtrless}} 0\,, \tag{3.7}$$

with $j=1,2,\ldots,\tilde{n}$, $\tilde{n}$ being the output layer dimension. Note that $\tilde{n}=1$ in binary classification, the case considered here.

Other non-linear discriminants can be obtained by applying other linear discriminants to the RBF output and thus, producing the non-linear DTC versions of the linear DTC discriminants seen in the previous chapter.

Part III

Results

4 Experiments

In this chapter, the performance of different linear and non-linear discriminants on several data sets is analyzed. These data sets are composed of artificial data with known distributions and real data with known and unknown distributions.

To determine which algorithm is better, the Test Set Accuracy (TSA) and the computational cost, in terms of training speed, and memory requirements, are considered.

Hypothesis test

The averages of the test accuracies obtained in the repetitions of each algorithm's measurement are compared following the Newman-Keuls hypothesis test with a confidence level of 95%, see Appendix D, for dependent samples (Pagano, 2010), *i.e.*, the same random partitions of the training and test sets are used for each algorithm evaluation.

All the algorithms have been written and run in Matlab by the authors without any external library nor routine. A Python version is also available.

4.1 Data sets

Except for the synthetic data sets used for the prior probability analysis in linear discriminants, the rest are real benchmark data sets taken from well known data bases.

	N	N_1	N_2	*Missing*	n
Two norm	7400	3703	3697	No	20
Breast cancer	683	444	239	No	10
Ionosphere	351	126	225	No	33
Heart disease	143	41	102	127	13
Vote	232	108	124	203	16
Sonar	208	111	97	No	60
Liver disorders	345	145	200	No	6
Skin segmentation	245057	50859	194198	No	3
SVM guide 1	7089	4000	3089	No	4
Cod RNA	331152	110384	220768	No	8
MNIST (0-1)	14780	6903	7877	No	400
MNIST (7-8)	14118	7293	6825	No	400
MNIST (4-5)	13137	6824	6313	No	400

Table 4.1: Data sets. $N/N_1/N_2$: Total/C_1/C_2 numbers of samples after remove missing; Missing: number of missing removed samples; n: number of features.

The main features of the data sets are shown in Table 4.1, which includes the number of samples after removing the missing samples and the number or these missing removed samples. Features are scaled into interval $[-1, 1]$, and the binary target values of the classes are $\{1, -1\}$.

Data sets

The data sets used are taken from the UCI database (Lichman, 2013) (Breast cancer, Ionosphere, Heart disease, Vote, Sonar, Liver disorders, Skin segmentation), the LIBSVM database (Chang and Lin, 2011) (SVM guide 1, Cod RNA), the Two norm database (Breiman, 1996) and the MNIST database (LeCun et al., 1998). Since the compared discriminants are binary classifiers, the MNIST problem is addressed through pair-wise digit comparisons. The pairs (0-1), (7-8) and (5-8) have been selected because they can be considered as low, medium and large difficulty, respectively.

For discussion of results purposes, the following data sets are considered data sets with large number of samples: Skin segmentation, SVM Guide 1, Cod RNA and MNIST. The first three are also considered data sets with a high ratio of samples per dimension.

Skin segmentation, *SVM Guide 1*, *Cod RNA*: $\frac{N}{n} \uparrow$; *MNIST*: $N \uparrow$

Each data set is randomly partitioned into a training set (70%) and a test set (30%). There are 100 runs of the algorithms in each training set, randomizing in each one the order of the samples. The same data set random partition is used in each comparison between algorithms, what is called a dependent samples situation (Pagano, 2010).

Training and test partitions

4.2 *Linear discriminants*

4.2.1 *Algorithms for comparision*

The linear discriminants used for comparison are the following:

- the DTC formulation of classical linear discriminants: MPDH-DTC, the direct solution of the minimax MPDH problem, alternative to the iterative MPM; Fisher-DTC, which adds an independent term to the direction provided by the Fisher criterion; and Scatter-DTC, based on the dispersion of the classes.

- a new DTC linear discriminant, called Quasi-Bayes-DTC, with very close accuracy to the optimal Bayes but lower computational cost.

- the linear MPM solution to the minimax MPDH problem (Lanckriet et al., 2002).

- the Bayes linear discriminant designed according to the Theoretical Method (TM) (Fukunaga, 1990).

4.2.2 *Prior behaviour*

The performance of the linear discriminants is analized in a prior probability swept P_1 of class C_1 in two synthetic distributions, uniform and Gaussian.

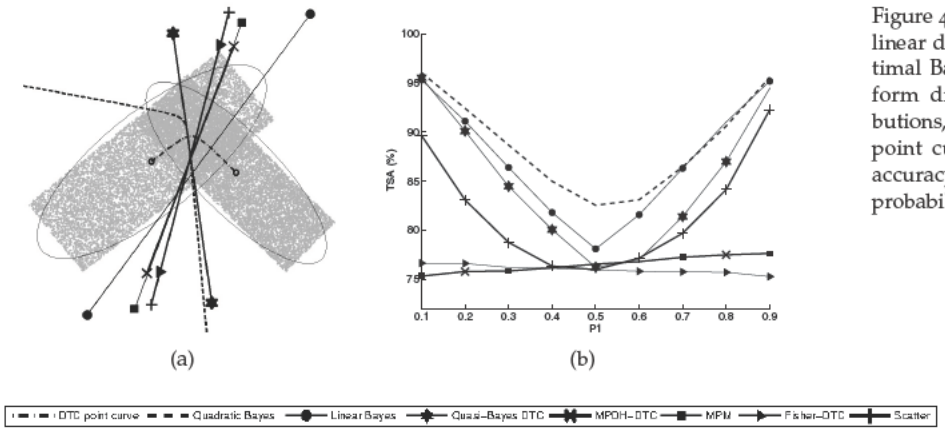


Figure 4.1: Prior behavior of the linear discriminants and the optimal Bayes with synthetic uniform distributions. (a) Distributions, discriminants and DTC point curve. (b) Classification accuracy (%) vs. C_1 class prior probability P_1.

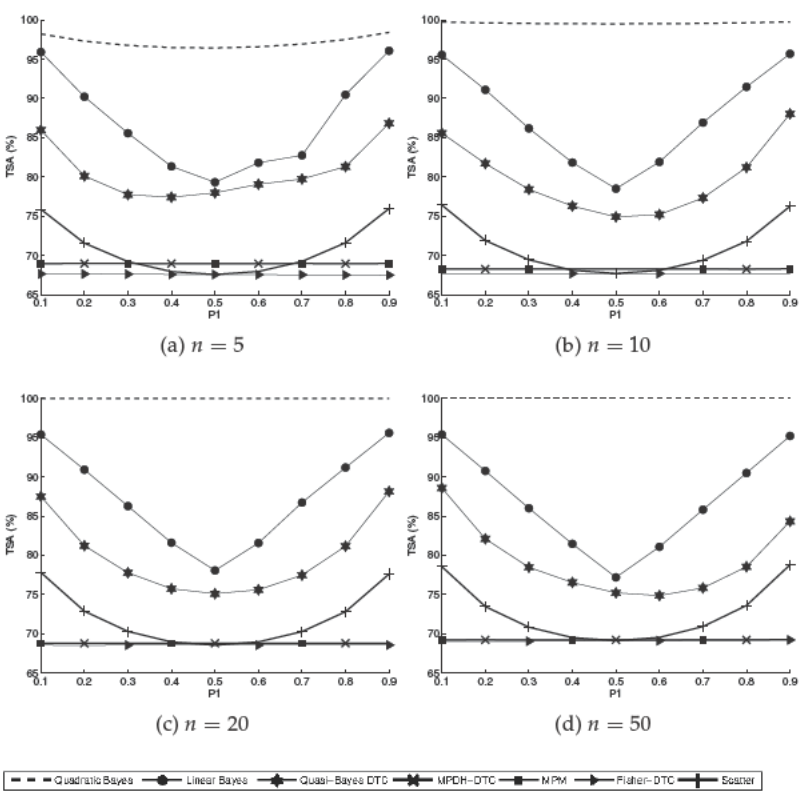


Figure 4.2: Prior behavior of the linear discriminants and the optimal Bayes with synthetic n-dimensional Gaussian distributions. Classification accuracy (%) vs. C_1 class prior probability P_1, for values of n equal to 5, 10, 20 and 50.

Figure 4.1 shows the binary classification of a pair of two-dimensional synthetic uniform distributions. On the left, the distributions and the ellipses of their covariance matrices, the linear discriminants, the DTC point curve, *i.e.*, the curve of all possible tangent points between

two ellipses centered in the means, and the Bayes optimal quadratic boundary, as a benchmark. The accuracy classification results as a function of the class C_1 prior probability P_1 are shown on the right.

The results for several n-dimensional synthetic Gaussian distributions are shown in Figure 4.2, where the performance of the discriminants is expressed in terms of the averaged classification accuracy for 100 runs.

Synthetic data sets

For each run, the mean vectors and the covariance matrices are uniformly random-generated. The means $\mathbf{m}_j$, j=1, 2, are in interval $[-1, 1]$. To generate the covariance matrices Σ_j, first, an element-by-element random matrix C is created in $[-2, 2]$; second, the transformation to get a definite-positive covariance matrix is $\Sigma = C \cdot C^T$, where T means transposed matrix.

In both uniform and Gaussian distributions, the discriminants only depend on the means and the covariance matrices, while the classification accuracy values are calculated generating 10^6 samples of each class.

Linear Bayes best accuracy, closely followed by Quasi-Bayes-DTC

In both distributions, linear Bayes, followed closely by Quasi-Bayes-DTC, provide larger accuracy than the rest of the linear discriminants. On the other hand, MPDH-DTC and MPM produce very similar results, both exhibit a minimax behavior in the sense of independence of prior probabilities. Although Fisher seems graphically similar to the MPDH-DTC and MPM discriminants, its behavior is not independent of the prior probabilities, *i.e.*, it is not minimax, the apparently flat result comes from the average of individual runs of the simulations.

In many cases, the relevant information about the problem, such as the class distributions or the prior probabilities, is unknown. Thus, the information obtained only from the data may be erroneous and the results poor. For this reason, the minimax solutions are used to minimize the maximum possible risk; in other words, they are methods whose worst solution (due to the unknown information) is the best possible.

4.2.3 *Benchmark problems*

Quasi-Bayes best accuracy

Test set accuracy (TSA). Table 4.2 shows the accuracy results for some real benchmark data sets. The best global results appear in bold numbers, while the best results within families are shown in italics, *i.e.*, the Bayesian family (comparing the Bayes linear discriminant with Quasi-Bayes-DTC) and the minimax family (comparing MPM with MPDH-DTC); in both cases following the already commented hypothesis test for means. According to this, Quasi-Bayes-DTC produces the best global result in more cases: Skin segmentation, SVM-guide-1, Cod-RNA and MNIST (0-1), tying with the Bayes linear discriminant in two of them and MPDH-DTC in one. It is also observed that the MPM algorithm is the best for Sonar. Linear Bayes,

MPM and MPDH-DTC produce the best result for MNIST (7-8). In general, it is not possible to establish a winner in the cases in which the number of samples per dimension is small, because it produces results with high dispersion and low reliability.

	Bayesian		Minimax			
	Linear-Bayes	Quasi-Bayes-DTC	MPM	MPDH-DTC	Fisher-DTC	Scatter-DTC
Two norm	97.8 ± 0.3	97.8 ± 0.3	97.8 ± 0.3	97.8 ± 0.3	97.8 ± 0.3	97.8 ± 0.3
Breast cancer	*97.0 ± 0.9*	95.0 ± 1.3	97.3 ± 0.9	97.2 ± 0.9	96.3 ± 1.4	97.2 ± 0.9
Ionosphere	*84.8 ± 3.3*	79.9 ± 3.9	*84.8 ± 2.9*	81.6 ± 3.3	79.0 ± 3.6	80.9 ± 3.3
Heart disease	83.4 ± 5.3	83.1 ± 5.2	81.2 ± 5.1	80.7 ± 5.1	80.9 ± 5.1	82.9 ± 5.0
Vote	96.0 ± 1.6	96.0 ± 1.6	96.0 ± 1.6	96.0 ± 1.6	96.1 ± 1.7	96.1 ± 1.8
Sonar	73.3 ± 5.8	71.9 ± 5.5	$\mathbf{77.1 \pm 5.2}$	73.6 ± 5.8	73.5 ± 5.9	73.2 ± 5.7
Liver disorders	*62.9 ± 4.6*	60.0 ± 5.3	60.9 ± 4.7	*62.9 ± 4.1*	63.1 ± 3.9	63.3 ± 3.9
Skin segmentation	93.3 ± 0.1	$\mathbf{93.6 \pm 0.1}$	93.4 ± 0.1	*93.5 ± 0.1*	91.8 ± 0.1	88.3 ± 0.1
SVM guide 1	$\mathbf{94.7 \pm 0.4}$	$\mathbf{94.7 \pm 0.4}$	93.5 ± 0.4	*94.2 ± 0.4*	86.0 ± 0.7	84.9 ± 0.7
Cod RNA	$\mathbf{95.1 \pm 0.0}$	$\mathbf{95.1 \pm 0.0}$	94.4 ± 0.0	*94.5 ± 0.1*	94.1 ± 0.1	95.0 ± 0.1
MNIST (0-1)	99.2 ± 0.2	$\mathbf{99.6 \pm 0.1}$	99.3 ± 0.1	$\mathbf{99.6 \pm 0.1}$	99.2 ± 0.2	99.3 ± 0.2
MNIST (7-8)	$\mathbf{98.2 \pm 0.2}$	98.0 ± 0.2	$\mathbf{98.1 \pm 0.2}$	$\mathbf{98.1 \pm 0.2}$	97.9 ± 0.2	98.0 ± 0.2
MNIST (5-8)	95.0 ± 0.3	94.3 ± 0.4	95.2 ± 0.3	95.1 ± 0.3	95.0 ± 0.3	95.0 ± 0.3

Table 4.2: Linear discriminants. Average of TSA (Test Set Accuracy) with standard deviation, in percent.

Another important issue is the quality of the characterization of the class distributions by their first two moments. For example, in the Liver disorders problem, the class distributions are not well characterized by their first two moments, because the third (skewness) and the forth (kurtosis) moments are high. That is the reason why Quasi-Bayes-DTC does not obtain a good result in this problem.

MPDH-DTC best minimax accuracy

Regarding the minimax discriminants, MPDH-DTC beats MPM in five cases (Liver disorders, Skin segmentation, SVM-guide-1, Cod-RNA and MNIST (0-1)) and ties in MNIST (7-8); MPM is better than MPDH-DTC in Ionosphere and Sonar, and there is no conclusion for the other four data sets, where results are very close. For the special case of the MNIST data set, MPDH-DTC is the method that provides more winning results. Note that the better performance of MPDH-DTC is obtained in the cases in which the number of samples per dimension is high; in these cases, the problem is well-defined by the data and a direct method such as MPDH-DTC provides more precise results than an iterative optimization method.

Being Quasi-Bayes the best algorithm in performance, it follows that it is also the best within the Bayesian family.

	Linear-Bayes	All DTCs	MPM
Two norm	20	8	**3**
Breast cancer	8	2	**1**
Ionosphere	40	9	**3**
Heart disease	10	**2**	**2**
Vote	10	3	**2**
Sonar	100	30	**10**
Liver disorders	8	2	**1**
Skin segmentation	40	**20**	**20**
SVM guide 1	6	**2**	3
Cod RNA	120	**60**	**60**
MNIST	500	240	**105**

Table 4.3: Linear discriminants. Average of training time in milliseconds.

Training time. Table 4.3 shows the training times in a x64-based 2.67 GHz processor without parallelization. Experience shows that MPM needs a low number of iterations to converge, exhibiting the better training speed. The Bayes linear discriminant is the slowest algorithm in training due to the search of the optimal parameter s (1.5). On the other hand, the computational costs in testing are similar for all the linear discriminants.

MPM best training time

4.2.4 *Discussion*

As a general conclusion, it can be established that, in the case that distributions are well represented by their first two moments and the relation between the number samples and their dimension is high, Quasi-Bayes-DTC is the best discriminant, even preferable to the Bayes linear discriminant due to its high performance and lower computational cost.

Conclusion: Quasi-Bayes winner, MPDH-DTC best minimax

In the case of needing a minimax solution, MPDH-DTC is better than MPM in the same circumstances as before, although with a slightly greater cost.

4.3 *Non-linear discriminants*

4.3.1 *Algorithms for comparision*

The non-linear discriminants used for comparison are:

- Gaussian RBFN with the following linear output discriminants:
  - the DTC formulation of classical linear discriminants: MPDH-DTC, Fisher-DTC and Scatter-DTC.
  - a new DTC linear discriminant, Quasi-Bayes-DTC.
  - the linear MPM solution to the minimax MPDH problem (Lanckriet et al., 2002).
  - the Bayes linear discriminant designed according to the Theoretical Method (TM) (Fukunaga, 1990).
- Gaussian-Kernel-MPM (Lanckriet et al., 2002), which provides a non-linear solution to the MPDH problem and serves for minimax comparison with RBF-MPM and RBF-MPDH-DTC due to the similarity of the Gaussian kernel with the Gaussian RBF nodes.

First, a comparison is made between all non-linear discriminants and then a particular comparison within the Bayesian family and the minimax family, paying special attention to RBF-MPDH-DTC and its alternative, Gaussian-Kernel-MPM.

	RBF-Linear-Bayes	RBF-Quasi-Bayes-DTC	Gaussian-Kernel-MPM	RBF-MPM	RBF-MPDH-DTC	RBF-Fisher-DTC	RBF-Scatter-DTC
Breast cancer	60 ± 14	34 ± 12	10 ± 2	21 ± 17	27 ± 14	42 ± 20	48 ± 26
Ionosphere	52 ± 15	66 ± 9	8 ± 1	70 ± 10	70 ± 13	56 ± 17	60 ± 17
Heart disease	15 ± 9	7 ± 3	4 ± 1	18 ± 17	35 ± 13	16 ± 6	15 ± 6
Vote	54 ± 16	32 ± 10	4 ± 1	47 ± 11	52 ± 7	60 ± 8	54 ± 15
Liver disorders	46 ± 16	6 ± 7	13 ± 1	44 ± 18	46 ± 12	46 ± 16	50 ± 21
Skin segmentat.	30 ± 4	25 ± 5	11 ± 1	76 ± 4	76 ± 4	31 ± 2	30 ± 2
SVM guide 1	70 ± 8	3 ± 0	14 ± 0	76 ± 5	76 ± 6	67 ± 7	63 ± 5
Cod RNA	64 ± 20	10 ± 0	13 ± 1	13 ± 2	34 ± 14	70 ± 6	42 ± 24

Table 4.4: Non-linear discriminants. Parameters of the algorithms. The average number of hidden nodes (m) in RBF algorithms and the average standard deviation (σ) in Gaussian-Kernel-MPM.

4.3.2 *Benchmark problems*

Parameters. The number of RBF kernels (scanned between 5 and 100) and the Gaussian-Kernel-MPM standard deviation (scanned between 1 and 15) are selected via 10-Fold CV for each run. The resulting values are shown in Table 4.4. The standard deviation of each RBF centroid is heuristically calculated as four times the mean of the Euclidean distances between the centroid and its nearest samples. The FSCL training to place the RBF centroids includes 100 epochs and an exponential decay of the learning parameter, see Appendix C.

Test set accuracy (TSA). In order to determine which algorithm is better, the test accuracy averages are compared in Table 4.5. The best global results appear in bold numbers, while the best results within the minimax and Bayesian families are shown in italics; following the already commented Newman-Keuls hypothesis test.

	Bayesian		Minimax				
	RBF-Linear-Bayes	RBF-Quasi-Bayes-DTC	Gaussian-Kernel-MPM	RBF-MPM	RBF-MPDH-DTC	RBF-Fisher-DTC	RBF-Scatter-DTC
Breast cancer	98.5 ± 0.6	98.5 ± 0.7	*98.7 ± 0.6*	97.9 ± 0.9	*98.8 ± 0.4*	99.1 ± 0.5	99.0 ± 0.5
Ionosphere	**97.1 ± 1.2**	**97.3 ± 1.3**	85.3 ± 2.2	63.9 ± 2.6	**97.5 ± 1.1**	**97.0 ± 1.2**	**97.0 ± 1.3**
Heart disease	**86.5 ± 4.3**	82.6 ± 4.7	**87.7 ± 3.6**	80.5 ± 6.4	**84.9 ± 3.8**	**85.6 ± 4.4**	**86.9 ± 3.5**
Vote	93.3 ± 1.3	93.3 ± 1.1	*93.3 ± 1.0*	85.8 ± 2.5	92.2 ± 1.6	93.4 ± 1.1	93.3 ± 1.1
Liver disorders	68.6 ± 3.6	61.3 ± 10.6	*64.5 ± 2.9*	52.7 ± 2.9	*66.5 ± 3.4*	66.1 ± 3.1	66.7 ± 4.3
Skin segmentation	**99.6 ± 0.1**	97.0 ± 0.6	**99.8 ± 0.0**	95.5 ± 0.8	**99.4 ± 0.1**	96.4 ± 8.9	97.3 ± 0.4
SVM guide 1	**96.6 ± 0.2**	89.9 ± 0.2	81.6 ± 0.8	91.0 ± 0.2	**96.4 ± 0.2**	**96.8 ± 0.1**	**96.6 ± 0.2**
Cod RNA	87.7 ± 0.3	88.0 ± 0.2	86.6 ± 0.3	80.8 ± 0.4	*87.5 ± 0.3*	87.2 ± 0.4	87.8 ± 0.4

Table 4.5: Non-linear discriminants. Average of TSA (Test Set Accuracy) with standard deviation, in percent.

Excluding RBF-MPM, the worst algorithm, the results are similar, specially for the first data sets, the problems with small number of samples per dimension. However, in the last three data sets, in which the number of samples per dimension is high, there is a small but

RBF-Linear-Bayes and RBF-MPDH-DTC best accuracy

clear accuracy advantage in choosing RBF-Linear-Bayes and RBF-MPDH-DTC. Considering the minimax results, the accuracy of RBF-MPDH-DTC is slightly better than that of Gaussian-Kernel-MPM. The poor results of RBF-MPM are due to the fact that the projection of the samples by the hidden layer increases the sparsity, which deteriorates the estimation of the covariance matrices, of which the MPM internal parameters are very dependent, accumulating errors in each iteration.

Presenting RBF-Linear-Bayes one of the best global performances, it follows that it is also the best within the Bayesian family.

Training Time. All simulations were run in one node of a computational cluster with 2.1GHz CPU and 64GB of RAM memory per node.

	RBF-Linear-Bayes	RBF-Quasi-Bayes-DTC	Gaussian-Kernel-MPM	RBF-MPM	RBF-MPDH-DTC	RBF-Fisher-DTC	RBF-Scatter-DTC
Two norm	1331	1364	7639	7572	1262	1745	1346
Breast cancer	23	24	5	631	33	26	25
Ionosphere	25	26	3	542	23	19	18
Heart disease	5	5	2	127	5	5	5
Vote	18	15	5	460	18	18	17
Sonar	7	8	3	327	14	11	12
Liver disorders	9	10	1	326	12	12	11
Skin segment.	755	625	12819	226029	754	642	713
SVM guide 1	1629	1019	7070	6721	1259	1295	1309
Cod RNA	1825	1606	20003	318353	1772	1974	2007

Table 4.6: Non-linear discriminants. Average of training time in milliseconds.

The results are shown in Table 4.6. Note the dramatic increase of the training time in the Gaussian-Kernel-MPM algorithm when the number of samples is high. The bad results of RBF-MPM are due to a poor convergence and the consequent iteration increment.

Gaussian-Kernel-MPM worst with high number of samples

Memory. While RBF-MPDH-DTC has to deal with m-dimensional square matrices, Gaussian-Kernel-MPM deals with N-dimensional square matrices, m being the number of the hidden nodes and N the number of samples. Since the number of samples is usually much larger than the number of hidden nodes, high numbers of samples will lead to high memory needs in Gaussian-Kernel-MPM. It is empirically demonstrated that, for more than 5×10^4 samples, Gaussian-Kernel-MPM requires a very large RAM memory, exceeding the limits of the machine described above.

4.3.3 *Discussion*

In general, and more specifically in problems which are well represented by their first two moments and in which the ratio of number

of samples per dimension is high, RBF-Linear-Bayes and MPDH-DTC present the best performance results, quite equal between them and the best option within their families. On the other hand, RBF-MPM exhibits a similar training time with the worst performance of all, while Gaussian-Kernel-MPM presents a similar performance, slightly worse, but with the worst training time.

4.4 *Computational cost*

The computational costs are shown in Table 4.7.

Linear			Non-linear			
MPM	All DTCs	Bayes	Gaussian-Kernel-MPM	RBF-MPM	RBF-All DTCs	RBF-Linear-Bayes
$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(N^3)$	$\mathcal{O}(m^3)$	$\mathcal{O}(m^3)$	$\mathcal{O}(m^3)$

Table 4.7: Computational cost of linear and non-linear algorithms. N: number of training samples; n: number of features; m: number of RBF nodes.

Linear discriminants. In the linear discriminants, although the computational cost of DTC appears to be slightly greater than the cost of MPM, it is necessary to consider that the value of MPM does not include the cost required for determining the optimal values of some internal parameters of the algorithm, such as the regularization parameter of the Hessian matrix and the stop-training tolerance parameter, which establishes that the summation of two variables, β_k and η_k, must be small enough (Lanckriet et al., 2002). Therefore, it is necessary to fit them by cross-validation. The cost of DTC is calculated considering the cost of solving the roots of the $2n$-grade Clark polynomial, $\mathcal{O}((2n)^3)$, and the cost of inverting n-dimensional square matrices, $\mathcal{O}(n^3)$; while MPM solves a n-order square-matrix linear system by Least Mean Squares (LMS) with a cost of $\mathcal{O}(n^3)$ for each iteration. The cost of the iterative algorithm to find the minimum Bayesian error in the Bayes linear discriminant includes the calculation of inverse square matrices of order n, as many as the number of steps Δs in which the search interval $[0, 1]$ of s is divided.

Linear: similar cost

Non-linear discriminants. For DTC linear algorithms, the main cost is solving the roots of the Clark polynomial of the order of the number of the sample features at the DTC input, so in the non-linear case that cost depends on m, the number of RBF nodes.

In Bayes, the cost is similar to DTC. This is because the computational load of solving the DTC Clark polynomial is practically the same than the load required by the iterative search for the minimum error in the Bayes method.

In the RBF algorithms (*i.e.*, all except Gaussian-Kernel-MPM), there is also the cost associated with the FSCL training of the centroids, which is not included because it is quadratic respect to the

number of features, and therefore less than the DTC polynomial cost of cubic order respect to the number of RBF nodes.

Gaussian-Kernel-MPM worst with high number of samples

On the other hand, Gaussian-Kernel-MPM solves a N-order square-matrix linear system by Least Mean Squares (LMS) with a $\mathcal{O}(N^3)$ cost per iteration. That is the reason why, in the non-linear case, if the number of samples is very high, the computational cost of Gaussian-Kernel-MPM dramatically raises compared with DTC. Moreover, Gaussian-Kernel-MPM presents an additional cost, like MPM, due to the internal parameters that need to be fixed by CV before training, *i. e.*, the regularization parameter of the Hessian and the stop-training tolerance parameter.

Theoretically, the RBF-MPM network presents a similar cost to RBF-MPDH-DTC since their output linear discriminants have a similar cost, as it is shown before. However, there exists a difference in the training time results due to bad convergence, already explained above.

4.5 Conclusions

	Linear				Non-linear				
	Bayesian		Minimax		Bayesian		Minimax		
	Linear-Bayes	Quasi-Bayes-DTC	MPM	MPDH-DTC	RBF-Linear-Bayes	RBF-Quasi-Bayes-DTC	Gaussian-Kernel-MPM	RBF-MPM	**RBF-MPDH-DTC**
Accuracy	✓	✓✓	✗✗✗	✗✗	✓✓✓	✗✗	✓✓	✗✗✗	✓✓✓
Computational cost	✓	✓✓	✓✓✓	✓✓	✓	✓	✗✗✗	✓	✓

✓: good performance. ✗: bad performance.

Table 4.8: Conclusions for the main algorithms.

In the linear algorithms, Quasi-Bayes-DTC is preferable, especially in problems with a high number of samples per dimension, even more than the Bayes linear discriminant, due to its high performance and lower computational cost, because it is a non-iterative solution; unless a minimax solution is required, in that case MPDH-DTC is preferred.

Regarding the non-linear algorithms, the Bayesian solution of RBF-Linear-Bayes and the minimax solution of RBF-MPDH-DTC globally perform almost equally, with a slightly better training time for RBF-MPDH-DTC, making it preferable because it is minimax too. The best minimax algorithm is RBF-MPDH-DTC, with better training time than Gaussian-Kernel-MPM and better accuracy than RBF-MPM.

In general, the training time of the linear algorithms is very low but with the disadvantage of a lower accuracy, except in problems which are linearly separable in its origin.

Considering all aspects, RBF-MPDH-DTC is the best choice due to its high accuracy with a competitive computational cost. Besides, with the advantage of providing a minimax solution, which can be useful in the case of the class-frequencies in the training, are not representative of the actual prior probabilities.

Part IV

Appendices

A

Classification error bounding

It is a classic statistical problem which consists of bounding the probability that a random variable belongs to a set (the misclassification set in this case) given the information of some of its moments. Notice that the solution of this problem is the bounding of the probability, independently of whether there exists a distribution which reaches this bound.

The problem is called (n, k, Ω)-*bound problem*, where n is the data dimension, k the number of known moments and Ω the set in which the probability that the random variable belongs to it is bounded.

There are different solutions which offer approaches based on semidefinite optimization (Bertsimas and Popescu, 2005) in order to calculate the bounds according to the values of n, k and Ω.

We only consider the two first moments $(n, 2, \mathbb{R}^n)$ for multidimensional problems. More moments may be considered, $k > 2$, and their corresponding formulations in order to bound the probability of error, but as the dimension of the data and the number of moments increases the calculation is more difficult. The two first moments may achieve a non-optimum bounding, but it is enough and easy to extract graphic conclusions with the probability ellipsoids.

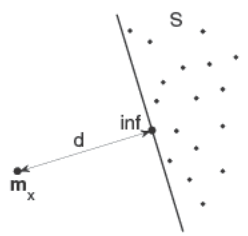


Figure A.1: Marshall. Bounding of probability that an aleatory vector $\mathbf{x}$ belongs to the set S given the two first moments. Mahalanobis distance d from the infimum of distances to the mean $\mathbf{m}_x$.

First, we define the quality of the found bounds. The term "best possible" or "tight" upper (and by analogy lower) bound γ of $P(\mathbf{x} \in S)$ is defined as follows

$$\gamma = \sup_{\mathbf{x} \sim \Pi} P(\mathbf{x} \in S) \,, \tag{A.1}$$

where $\Pi = (M_1, M_2, \ldots, M_k)^T$ is a sequence of k feasible moments, $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ defined in $\Omega \subseteq \mathbb{R}^n$ has a feasible distribution in Π, written as $\mathbf{x} \sim \Pi$, and finally $S \subseteq \Omega$ is a semi-algebraic set, i.e. defined in terms of polynomial inequalities.

Note that a bound can be tight without necessarily being exactly achievable, but only asymptotically.

Second, in order to find bounds in the univariate case, the inequalities of Markov $(1, 1, \Omega)$, Chebyshev $(1, 2, \Omega)$ and Chernoff are used if the first moment, the two first moments and all moments (i.e. the

generative function) of a random variable are known, respectively. They are are feasible but not necessarily optimal solutions or tight bounds to the (n, k, Ω) bound problem.

Marshall and Olkin (1960) show it is possible to explicitly calculate the probability bounds given the two first moments $(n, 2, \mathbb{R}^n)$ generalizing the Chebyshev inequality for the multivariate case, the case in which we are interested

$$\sup_{\mathbf{x}\sim(\mathbf{m}_x,\Sigma_x)} P\{\mathbf{x} \in S\} = \frac{1}{1+d^2} \ , \tag{A.2}$$

with

$$d^2 = \inf_{\mathbf{x}\in S} (\mathbf{x}-\mathbf{m}_x)^T \Sigma_x^{-1} (\mathbf{x}-\mathbf{m}_x) \ , \tag{A.3}$$

where $\mathbf{x}$ is a random vector, S is a convex half-space, *i.e.*, separates space into two convex sets, one of them is itself. The Mahalanobis distance d is a multi-dimensional measuring of how many standard deviations away is a point from the mean. Since the covariance matrix of the distribution is taken into account, the distances of Mahalanobis correspond to ellipsoidal level surfaces, instead of spherical level surfaces of euclidean distances, in which the covariance matrix is the identity matrix. The boundary of S, because it is a convex set, is tangent to a determinate level surface and the tangent point is the infimum of all Mahalanobis distances from the mean to all points of S. Notice that the infimum is always unique if it exists. The probability of being part of S decreases with the distance d because the greater distance from the mean to the boundary of S the smaller region S and less points fall in it. So, given a fixed boundary, the supremum of the probability is reached with the infimum of the distances. The obtained solution is valid for all possible distributions of $\mathbf{x}$ given the same two first moments, without assuming gaussianity. A toy example is shown in Fig. A.1.

In binary classification, S corresponds to the misclassification region. The upper bound of the misclassification probability or error probability for new samples, *i.e.*, they fall in region S, is described by (A.2). The aim will be to minimize that misclassification probability.

	Accuracy		
Distributions	Total	C_1	C_2
Gaussian	94.79	94.8	94.8
t	95.57	95.5	95.7
Uniform	98.68	98.6	98.7

Table A.1: Toy example: success accuracy (%) in distributions, with the same mean vector and covariance matrix per class, given the accuracy bound $\varepsilon = 75\%$. Number of samples $N = 10^5$

B

Minimax criterion

Minimax means maximizing first and minimizing after, or vice versa, minimizing first and maximizing after, due to the principle of duality.

The aim of minimax is minimizing the worst case or maximum risk. In classification, it consists of searching the discriminant which minimizes the maximum error of each possible classifier. Thus, there are two risks to minimize: first, the classification error produced by the shift of the prior probabilities regarding the training; and second, the unknowledge of the data distribution and the possible assumptions about it and, as a special case, its prior probabilities.

Figure B.1 shows in the concave solid curve the misclassification probability of the optimal Bayes discriminant for each prior probability value of the class C_1. The points A and B are two examples of the optimal Bayes misclassification probability for two given prior probabilities. Once the the Bayes classifier is trained for a given prior probability, *e.g.*, point B, if that prior changes, the misclassification error is linear with the change of the prior and tangent with the Bayes optimal curve in the starting point, as shown in the dashed line and the points B^- and B^+ for two given shifts. The minimax solution corresponds to the point A, which is the worst-case of the Bayes misclassification probability, *i.e.*, the greatest error, but also is the minimum worst-case, for any shift of the prior, *e.g.*, lower than B^+. Therefore, the minimax solution minimizes the worst-case. A consequence of the minimax solution is the independence of the prior probabilities, as shown in the horizontal line.

A consequence of the independence of the prior probabilities in the minimax solution is that the error (or success) probability of each class is equal. The total error it is a weighting of each class error multiplied by the prior probabilities. Thus, it is equivalent that both class errors are the same and independent from the prior probabilities in the total error.

Toy example:

Solve maximizing the sum, subject to a restriction, being each unknown the smallest possible.

$$\max_{min\ x,y,z} \quad x+y+z$$
$$s.t. \quad x+y+z \leq 16$$

The solution is a combination of 5, 5 and 6. If it was not minimax, the solution could be 14, 1 and 1.

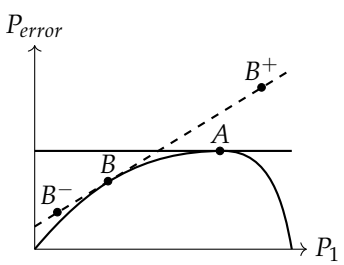


Figure B.1: Minimax behaviour with prior probabilities. Horizontal: class C_1 prior probability. Vertical: misclassification probability.

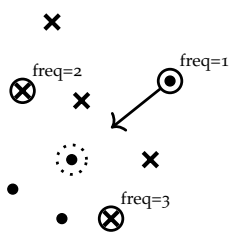
C

Frequency Sensitive Competitive Learning (FSCL)

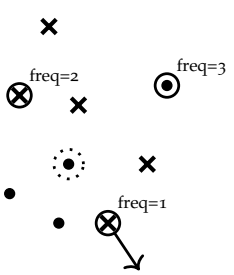
The centroids are trained with the Frequency Sensitive Competitive Learning (FSCL) algorithm by Ahalt et al. (1990). This is a competitive learning procedure that proposes a new distance measure by multiplying the number of times a neuron wins to the original distance function. In this way, a neuron –in our case, centroids of the RBF– which frequently wins in the competition for getting a sample will have a lower probability to win next time. It is proven that this mechanism converges to positions of the neurons that maximize the entropy, which is of great importance because the selected centroids will appropriately represent the population of training samples, avoiding the risks of under/over-representation of parts of the population. It also allows to estimate the Gaussian standard deviations averaging the distance from each centroid to its nearest samples. The number of centroids, which is also the dimension of the hidden layer, is selected by Cross-Validation (CV), hence it is different for each algorithm and data set.

FSCL competitive learning:

- For each sample:
  1. Search of the nearest centroid to the sample
     - distance multiplied by the winning frequency
  2. Movement of the winner centroid
  3. Increment of the winning frequency



(a) Same class



(b) Different class

Figure C.1: FSCL algorithm. Movement of centroids. For each sample, the nearest centroid (distance multiplied by the frequency of winning) moves towards the sample if they are the same class and in the opposite direction if they are different classes. The samples are the crosses and points, the centroids are the circles and the sample analyzed is inside the dotted circle.

C.1 Algorithm

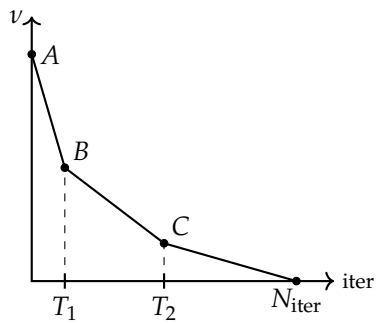
Algorithm 3: Frequency Sensitive Competitive Learning (FSCL)

Data: $X_{[N\times n]}$ labeled patterns, N_c: number of centroids
Result: $C_{[N_c\times n]}$ centroids

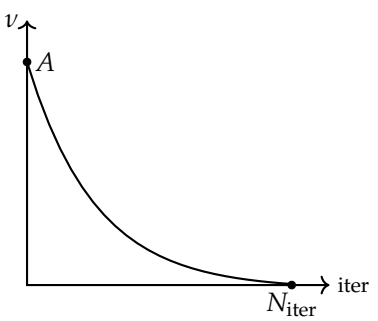
$N_e = 100$ `% Number of epochs`
$N_{\text{iter}} = N \cdot N_e$ `% Number of iterations`

`% ν Learning parameter calculation`
$A = 0.3$
if *linear decay* **then**
  $B = 0.15;\ C = 0.05$
  $T_1 = 0.125 \cdot N_{\text{iter}};\ T_2 = 0.5 \cdot N_{\text{iter}};\ T_3 = N_{\text{iter}}$
  $\nu[1:T_1] = -\frac{A-B}{T_1}[1:T_1] + A$
  $\nu[T_1:T_2] = -\frac{B-C}{T_2-T_1}\left([T_1:T_2] - T_1\right) + B$
  $\nu[T_2:T_3] = -\frac{C}{T_3-T_2}\left([T_2:T_3] - T_2\right) + C$
else if *exponential decay* **then**
  $\tau = 2/N_{\text{iter}}$ `% Time constant`
  $\nu = \frac{A}{e^{-\tau} - e^{-\tau\cdot N_{\text{iter}}}}\left(e^{-[1:N_{\text{iter}}]} - e^{-\tau\cdot N_{\text{iter}}}\right)$

`% ν Centroid calculation`
$\text{freq}_m \leftarrow 0,\ m = 1,2,\ldots,N_c$ `% Frequency initializations`
$C_m \leftarrow \text{rand}$ `% Random Position initializations`
for $e = 1$ **to** N_e **do**
  foreach X_i, i=1,2,...,N **do**
    `% j: index of the nearest centroid to` X_i
    $j \leftarrow \arg\min_m\{\text{freq}_m \cdot \|X_i - C_m\|\}$
    `% Shift of the winner centroid` C_j
    $k = e + i$
    if X_i *same class as* C_j **then**
      $C_j \leftarrow C_j + \nu_k\left(X_i - C_j\right)$
    else
      $C_j \leftarrow C_j - \nu_k\left(X_i - C_j\right)$
    `% Winner frequency update` $\text{freq}_j \leftarrow \text{freq}_j + 1$



(a) Piecewise linear decay



(b) Exponential decay

Figure C.2: FSCL algorithm. Learning parameter ν decay

D

Hypothesis Test

The aim of a scientific experiment is the validation of a hypothesis. The most common hypothesis to prove is whether the performance of one method is better than other.

The first step of the methodology consists of selecting a representative sample of the population to which the hypothesis will be generalized. The way to validate the hypothesis is to divide it into two: the alternative hypothesis (H_1) and the null hypothesis (H_0). The alternative hypothesis explains the difference of results in the different methods is due to a real factor, *i.e.*, the cause is the independent variable. On the contrary, the null hypothesis claim to that difference is due to only chance factors (Pagano, 2010).

The procedure of hypothesis testing in statistical inference is the following: First, let us assume the null hypothesis is true and evaluate that the chance-alone probability is the cause of the results difference; second, compare that obtained probability with a threshold level called critical probability (α). If it is greater than the critical probability, the null hypothesis is retained concluding the difference of results are due to only chance, *i.e.*, both methods produce the same results. On the contrary, if it is lower, the null hypothesis is rejected and the alternative hypothesis is accepted, *i.e.*, the difference of results are due to the independent variable, which is different in both methods, and then one is better than the other.

There exists two type of errors derived from incorrectly deciding, see Table D.1: The Type I error, as a result of deciding to reject the null hypothesis when the null hypothesis is true; and the Type II error when a false null hypothesis is retained.

The critical probability level (α) is the limit of the Type I error, so decreasing α decreases the Type I error, but increases the Type II error. In manufacture and presenting new scientific discoveries, it is better to diminish the Type I error by selecting a low value of α, typically 0.01 or 0.05. Nevertheless, in the case of the exploration of new possibilities, it is more important to reduce the Type II error, hence higher values of α are considered, like 0.10 or even 0.20.

Sample or group. Set of individuals evaluated in a certain **condition**, *e.g.*, an algorithm, method or a placebo test

Individual or subject. (*e.g.*, a random partition of a dataset)

Dependent variable. The effect of the experiment, the measures

Independent variable. The cause (not random) of the dependent variable

Alternative hypothesis. Asserts that the differences in results are caused by the independent variable

Null hypothesis. Logical opposite to the alternative hypothesis. Asserts that the independent variable has no effect on the dependent variable, the cause is chance alone

Critical probability level. The null hypothesis is rejected if the probability of chance alone being the cause of the results is equal or less than this level

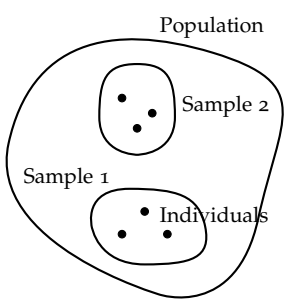


Figure D.1: Example of a population and 2 samples with $N=3$ individuals.

The power measures the sensibility of the experiment to detect a real effect, the effect that the independent variable produces in the dependent variable. It is also the probability of rejecting the null hypothesis correctly, hence it is desirable that it has a high value, as close to 1 as possible; however, it is difficult to see higher values than 0.8, 0.4 to 0.60 is common. The power depends of the size of the real effect, the higher the contribution of the independent variable, the greater the power. Increasing the sample size (N) also increases power.

$$\beta = 1 - \text{power} \, . \tag{D.1}$$

Completing the hypothesis testing method, the first step is the calculating of the statistic, *e.g.*, the mean of the accuracy performances used to compare algorithms or methods; second, assuming the null hypothesis is null, calculate the probability of chance alone of the obtained results with the sampling distribution of the statistic; and finally, compare that probability of chance alone with the critical probability to reject or not the null hypothesis.

	Reality	
Decision	H_0 true	H_0 false
Retain H_0	0	Type II (β)
Reject H_0	Type I (α)	0

Table D.1: Errors in accepting and rejecting H_0 incorrectly

Dependence of errors

$\alpha \downarrow \begin{cases} \text{Type I error} \downarrow \\ \text{Type II error } (\beta) \uparrow \end{cases}$

$\left.\begin{array}{l} N \uparrow \\ \text{real effect size} \uparrow \end{array}\right\}$ Power $\uparrow$, $\beta \downarrow$

D.1 *Statistical tests*

The sampling distribution of a statistic give the probabilities of the values the statistic can take if they are caused by chance alone, it is the previous step to the comparison with the critical probability α, as shown before.

The null-hypothesis population is an actual or theoretical set resulting of doing the experiment in the entire population in the case of the independent variable has no effect. It is used to test the null-hypothesis H_0 taking samples of N individuals, as many as combinations of N individuals exist. The statistic, *e.g.*, the mean, is evaluated in each sample, and the set of all of these statistics form the null-hypothesis population. Depending on the chosen statistic and the sampling distribution characterization, there are different statistical tests, the most important ones are shown below.

Hypothesis testing

1. Statistic calculation (*e.g.*, mean)
2. Sampling distribution calculation (probability of chance alone)
3. P. chance $\begin{cases} > \alpha \Rightarrow \text{retain } H_0 \\ \leq \alpha \Rightarrow \text{reject } H_0 \\ \qquad (\text{accept } H_1) \end{cases}$

D.2 *The normal deviate z-test*

In this test, the statistic is the mean and the sampling distribution of the statistic is a normal distribution.

The requisites for using this test are: The parameters of the null-hypothesis population (μ, σ) are known; the sampling distribution of the mean is normal, that occurs when the null-hypothesis distribution is normal or when null-hypothesis distribution is close to normal and the number of observations of the sample is$N \geq 30$, by the Central Limit Theorem; and there is a single sample, so the null-hypothesis asserts that the sample is a random sample of the null-hypothesis

z-Test requirements

- H_0 Pop. (μ, σ) known
- Samp. dist. of means ($\mu_{\bar{X}}$, $\sigma_{\bar{X}}$) is normal $\begin{cases} H_0 \text{ pop. is normal } \textit{or} \\ N \geq 30 \end{cases}$
- Single sample

population and the alternative hypothesis asserts the opposite.

The parameters of the sampling distribution are the mean ($\mu_{\bar{X}}$) and the standard deviation ($\sigma_{\bar{X}}$), with the next relation with the null-hypothesis population parameters:

$$\mu_{\bar{X}} = \mu \,, \tag{D.2a}$$

$$\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{N}} \,, \tag{D.2b}$$

D.3 Student's t test

Only the mean is affected by the independent variable, not the variance.

D.3.1 Single sample

It is similar to the z-test, but here it is presented as a more common case, a null-hypothesis population of which the mean is known but not the standard deviation, that is the reason why the normal distribution is not used as the sampling distribution; instead, the t-distribution is used as the sample distribution of the statistic, the mean. Since σ is unknown, it is estimated from the sample and called s. Thus, the parameters of the sampling distribution are as follow:

Single sample t-Test requirements

- H_0 Pop. (μ, ~~σ~~) known
- Samp. dist. of means ($\mu_{\bar{X}}$, $\sigma_{\bar{X}}$) is normal $\begin{cases} H_0 \text{ pop. is normal } \textit{or} \\ N \geq 30 \end{cases}$
- Single sample

$$\mu_{\bar{X}} = \mu \,, \tag{D.3a}$$

$$s_{\bar{X}} = \frac{s}{\sqrt{N}} \,, \tag{D.3b}$$

$s_{\bar{X}}$ being an estimate of $\sigma_{\bar{X}}$.

The t-distribution looks like the normal distribution with the difference that they are a family of curves which depend on the sample size N. More precisely, the t-distribution varies with the degrees of freedom (df) of the sample, *i.e.*, the number of individuals that are free to vary in calculating the statistic.

$$df = N - 1 \,, \tag{D.4}$$

if $df = \infty$ the t-distribution is equal to the normal distribution.

D.3.2 Two samples

This is the case called two-condition, two-groups, or two-samples experiment, understanding that the same set of individuals evaluated in two different conditions constitutes two different samples, because the populations they come from are different.

Two correlated groups t-Test req.

- H_0 pop. param. ($\mu_D = 0$, ~~σ_D~~)
- Sampling dist. of mean of diferences (μ_D, σ_D) is normal $\begin{cases} H_0 \text{ pop. is normal } \textit{or} \\ N \geq 30 \end{cases}$

Correlated groups. In the correlated groups, each subject is evaluated in two different conditions. It is also possible to use different subjects matched in pairs that share the same characteristics, *e.g.*, age or

gender and so forth. The null-hypothesis claims that the difference of results in the conditions is a random sample from a population of differences with zero mean, *i.e.*, there is no difference between results. Hence, the parameters of the sampling distribution of the statistic, which is the mean of the differences of the results, are $\mu_D = 0$ and σ_D (unknown, but not needed for the t-test).

Independent groups. The subjects are randomly selected from the subject population. There are two null-hypothesis populations. The statistic is the difference of the means of each group.

The homogeneity of variance consists of the assumption that the variances of the two populations are equal $\sigma_1^2 = \sigma_2^2$. Nevertheless, the robustness of the t-test makes it relatively insensitive to violations of the normality and the homogeneity of variance. But if the variances are very different may be the independent variable has not equal effect in both populations.

One degree of freedom is lost each time a standard deviation is estimated

$$df = N - 2 \,, \tag{D.5}$$

where $N = n_1 + n_2$, n_1 and n_2 being the sizes of each sample.

Two indep. groups t-Test req.

- Two H_0 pops. param. ($\mu_1 - \mu 2 = 0$, ~~$\mu_1 - \mu_2 = 0$~~)
- Sampling distribution of difference of means $\bar{X}_1 - \bar{X}_2$ is normal $\begin{cases} H_0 \text{ pops. are normal } \textit{or} \\ n_1 \geq 30 \;\&\; n_2 \geq 30 \end{cases}$
- Var. homogeneity $\sigma_1^2 = \sigma_2^2$

Correlated vs. independent Even though in each problem a choice or another may be more natural, it is necessary to consider some aspects. In the correlated groups, the variability of the difference of results is lower, which increases power. While the degrees of freedom in the independent groups are greater, which diminishes the critical value of t.

Correlated groups
sample variability $\downarrow \Rightarrow$ power $\uparrow$

Independent groups
$df \uparrow \Rightarrow t_{\text{crit}} \downarrow \Rightarrow$ Type I error $\uparrow$

D.4 Multiple samples

Unlike the previous test, the mean is not used as statistic but the variance, with the F-distribution as the sampling distribution, which is a ratio between two independent estimates of the population variance. It is appropriate for analysis of more than two samples and the conditions can be the effect of different independent variables or the range the independent variable. The reason for not making multiple comparisons with the previous test for two samples between pairs of conditions is that multiple t or z evaluations increase the Type I error. Even though the F-test analyzes variance, it allows to make one overall between the means of the groups avoiding the increasing the probability of Type I error. It is used both in independent and repeated measures groups and when two or more factors are investigated.

F-test avoids the increase of Type I error in multiple comparisons but not specifies which condition is better

Like the t-test, it is assumed that only the mean is affected by the independent variable, not the variance.

The null-hypothesis claims that all the conditions have the same effect on the dependent variable, while the alternative hypothesis, which is always non-directional, states than one or more conditions have different effects.

The F-test is the ratio between two variance estimations of the null-hypothesis population: The between-groups variance and the withing-groups variance. The higher the effect of the independent variable, the higher the between-groups variance, while the withing-groups variance remains the same because the each group receive the same level of independent variable. Hence, the higher the effect of the independent variable, the higher the F value and the more probability of reject the null-hypothesis.

$$F_{\text{obt}} = \frac{s_B^2}{s_W^2} \,, \tag{D.6}$$

s_B^2 being the between-groups variance and s_W^2 the withing-groups variance. Note if $F_{\text{obt}} \leq 1$ chance alone is the explanation and the null-hypothesis is retained. In the case of two independent groups, the t-test and the F-test are related by $t^2 = F$.

F-Test requirements

- Populations from the samples are normal
- Variance homogeneity $\sigma_1^2 = \sigma_2^2 = \ldots = \sigma_k^2$

The assumptions of the variance analysis for k groups are similar to those of the t-test for independent groups. Like the t-test, the F-test is robust. It is affected little by populations that are close to normal and it is relatively insensitive to violations of variance homogeneity. As in the previous tests, the power increases with N, it is greater for large effects of the independent variable and the lower the sample variability, the greater the power to detect the real effect.

Power

$N \uparrow$
real effect size $\uparrow$
sample variability $s_W^2 \uparrow$
} Power $\uparrow$

When multiple comparisons are made, it is necessary to correct the increment of the probability of the Type I error. There are two main methods that maintain the Type I error rate at α while making all possible comparisons: The HSD (Honestly Significant Difference) by Tuckey and the Newman-Keuls test. These methods use the Q or Studentized distributions.

D.5 Newman-Keuls test

Newman-Keuls avoids the increase of Type I error in multiple comparisons like F-test but specifies which condition is better

The difference with the F-test is that its alternative hypothesis asserts that one or more conditions are different with the other, but does not specify which ones because it does not make multiple comparisons. Instead, the Newman-Keuls tests make the multiple comparisons establishing the winner or winners in a rank-ordered mean list.

This is an *a posteriori* test, in which the comparisons are not planned in advanced, which allows to correct the inflated probabilities of Type I error when doing multiple comparisons. The Newman-Keuls method maintains the Type I error at α for each comparison, unlike the HSD method, which maintains the error for the full set of possible comparisons. The Newman-Keuls test is more

powerful because it presents a higher Type I error for the experiment but a lower Type II error. Conversely, HSD is more conservative.

			Null-hypothesis population	
	Test	Statistic	Information	Sampling distribution
One sample	z-test	mean	μ, σ	Normal
One sample	t-test	mean	μ	t-dist.
Two correlated samples	t-test	mean of differences	$\mu_D = 0$	t-dist.
Two independent samples	t-test	difference of means	$\mu_1 - \mu_2 = 0$	t-dist.
Multiple samples*	F-test	variance	$\mu_1 = \mu_2 = \ldots = \mu_k$	F-dist.
Mult. samp.* and comparisons	Newman-Keuls	variance	$\mu_1 = \mu_2 = \ldots = \mu_k$	Q-dist.

Table D.2: Summary of the hypothesis tests. *$k > 2$ samples.

Bibliography

Ahalt, S. C., Krishnamurthy, A. K., Chen, P. and Melton, D. E. (1990) Competitive learning algorithms for vector quantization. *Neural Netw.*, 3(3):277–290. doi: 10.1016/0893-6080(90)90071-R.

Bengio, S., Mariéthoz, J. and Keller, M. (2005) The expected performance curve. In *International Conference on Machine Learning, ICML, Workshop on ROC Analysis in Machine Learning*, Idiap-RR-85-2003. Bonn, Germany.

Bertsimas, D. and Popescu, I. (2005) Optimal inequalities in probability theory: A convex optimization approach. *SIAM J. Optim.*, 15(3):780–804. doi: 10.1137/S1052623401399903.

Bishop, C. M. (1995) *Neural Networks for Pattern Recognition*. Oxford Univ. Press, UK. ISBN 0-19-853864-2.

Bishop, C. M. (2006) *Pattern Recognition and Machine Learning*. Springer, Cambridge, UK. ISBN 978-0387-31073-2.

Boyd, S. and Vandenberghe, L. (2004) *Convex Optimization*. Cambridge Univ. Press, UK. ISBN 978-0-521-83378-3.

Breiman, L. (1996) Bias, variance and arcing classifiers. Technical Report 460, Statistics Department, University of California. http://www.cs.toronto.edu/~delve/data/twonorm/desc.html.

Chang, C.-C. and Lin, C.-J. (2011) LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27. http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

Clark, M. P. (1995) On the resolvability of normally distributed vector parameter estimates. *IEEE Trans. Signal Process.*, 43(12):2975–2981. doi: 10.1109/78.476441.

Doersch, C. (2016) Tutorial on variational autoencoders. *arXiv e-prints*, arXiv:1606.05908.

Duda, R. O., Hart, P. E. and Stork, D. G. (2000) *Pattern Classification*. Wiley–Interscience, New York, NY, 2nd edition. ISBN 978-0471056690.

Fisher, A. (1923) *The Mathematical Theory of Probabilities*. Macmillan, New York, NY.

Fukunaga, K. (1990) *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, CA, 2nd edition. ISBN 978-0-08-047865-4.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014) Generative adversarial networks. *arXiv e-prints*, arXiv:1406.2661.

Haykin, S. (2009) *Neural Networks and Learning Machines*. Pearson-Prentice Hall, New York, NY, 3rd edition. ISBN 978-0-13-147139-9.

Huang, G.-B., Zhu, Q.-Y. and Siew, C.-K. (2006) Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1):489–501. doi: 10.1016/j.neucom.2005.12.126.

Huang, K., Yang, H., King, I., Lyu, M. R. and Chan, L. (2004) The minimum error minimax probability machine. *J. Mach. Learn. Res.*, 5:1253–1286. www.jmlr.org/papers/volume5/huang04a/huang04a.pdf.

Jollife, I. T. (2002) *Principal Component Analysis*. Springer-Verlag, New York, NY, 2nd edition. ISBN 978-0-387-22440-4. doi: 10.1007/b98835.

Lanckriet, G. R., El Ghaoui, L., Bhattacharyya, C. and Jordan, M. I. (2002) A robust minimax approach to classification. *J. Mach. Learn. Res.*, 3:555–582. www.jmlr.org/papers/volume3/lanckriet02a/lanckriet02a.pdf.

LeCun, Y., Cortes, C. and Burges, C. J. (1998) The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist.

Lichman, M. (2013) UCI Machine Learning Repository. http://archive.ics.uci.edu/ml.

Makhzani, A., Shlens, J., Jaitly, N. and Goodfellow, I. J. (2015) Adversarial autoencoders. *arXiv e-prints*, page arXiv:1511.05644.

Marshall, A. W. and Olkin, I. (1960) Multivariate Chebyshev inequalities. *Ann. Math. Statist.*, 31(4):1001–1014. doi: 10.1214/aoms/1177705673.

Martínez-García, J.-A. and Sancho-Gómez, J.-L. (2018) Performance analysis of No-Propagation and ELM algorithms in classification. *Neural Comput. Appl.* doi: 10.1007/s00521-018-3353-0. http://rdcu.be/E68l.

Martínez-García, J.-A., Sancho-Gómez, J.-L., Sánchez-Morales, A. and Figueiras-Vidal, A. R. (2019) Designing non-linear minimax and related discriminants by disjoint tangent configurations applied to RBF networks. *Neurocomputing*. Unpublished. In revision.

Pagano, R. R. (2010) *Understanding Statistics in the Behavioral Sciences*. Wadsworth Cengage Learning, Belmont, CA, 9th edition. ISBN 978-0-495-59652-3.

Peterson, D. W. and Mattson, R. L. (1966) A method of finding linear discriminant functions for a class of performance criteria. *IEEE Trans. Inf. Theory*, 12(3):380–387. doi: 10.1109/TIT.1966.1053913.

Sánchez-Morales, A., Sancho-Gómez, J.-L., Martínez-García, J.-A. and Figueiras-Vidal, A. R. (2019) Improving deep learning performance with missing values via deletion and compensation. *Neural Comput. Appl.* doi: 10.1007/s00521-019-04013-2.

Sancho-Gómez, J.-L., Martínez-García, J.-A., Ahalt, S. C. and Figueiras-Vidal, A. R. (2018) Linear discriminants described by disjoint tangent configurations. *Neurocomputing*, 316:345–356. doi: 10.1016/j.neucom.2018.08.010.

Schölkopf, B. H. and Smola, A. J. (2001) *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA. ISBN 978-0262194754.

Sebastiani, F. (2002) Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47. doi: 10.1145/505282.505283.

Vapnik, V. (1995) *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, NY. ISBN 978-1475732641.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. and Manzagol, P.-A. (2010) Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408. http://dl.acm.org/citation.cfm?id=1756006.1953039.

Widrow, B., Greenblatt, A., Kim, Y. and Park, D. (2013) The No-Prop algorithm: A new learning algorithm for multilayer neural networks. *Neural Netw.*, 37:182–188. doi: 10.1016/j.neunet.2012.09.020.

Index of Figures

Index of Tables


UPCT
INDOCT
ESCUELA
INTERNACIONAL DE
DOCTORADO