# Índice general

# Parte I

# Matemáticas

# Capítulo 1

# Análisis convexo

El análisis convexo es un campo de estudio fundamental para muchos problemas de optimización. En él, se estudian los conjuntos, funciones y problemas convexos. Las funciones convexas presentan propiedades muy útiles en tareas de optimización, y permiten construir herramientas para resolver numerosos tipos de problemas de optimización convexos.

El análisis convexo es una rama del análisis muy desarrollada. Sobre este tema se han desarrollado capítulos y hasta libros completos [3, 28, 10]. En este trabajo nos centraremos en una parte muy reducida del análisis convexo, en la cual presentaremos algunas propiedades geométricas de los conjuntos convexos, destacando el teorema de la proyección convexa, y recordaremos algunas de las propiedades más importantes de las funciones convexas, que serán de utilidad más adelante. Por último, se presentará la formulación de los problemas de optimización, centrándonos en aquellos convexos, y proporcionando herramientas básicas para resolverlos.

## 1.1. Conjuntos convexos

### 1.1.1. Definición y propiedades

Comenzamos recordando el concepto de conjunto convexo y algunas de sus principales propiedades. En este tema trabajaremos en $\mathbb{R}^d$ con la estructura de espacio de Hilbert, donde el producto escalar lo notaremos por $\langle \cdot, \cdot \rangle$.

**Definición 1.1.1.** Dados $x_1, x_2 \in \mathbb{R}^d$, se define el *segmento* que une $x_1$ y $x_2$ como $[x_1, x_2] = \{(1-\lambda)x_1 + \lambda x_2 \colon \lambda \in [0,1]\} = \{x_1 + \lambda(x_2 - x_1) \colon \lambda \in [0,1]\}$.

**Definición 1.1.2.** Un subconjunto $K \subset \mathbb{R}^d$ se dice que es convexo si, para cualesquiera dos puntos de $K$, el segmento que los une está contenido en $K$, esto es,

$$x_1, x_2 \in K \implies [x_1, x_2] \subset K.$$

Son inmediatos los siguientes ejemplos y propiedades sobre los conjuntos convexos.

a) Los subespacios vectoriales son convexos.

b) Los semiespacios $\{x \in \mathbb{R}^d \colon T(x) < \alpha\}$ (resp. $>$, $\leq$, $\geq$), donde $T \colon \mathbb{R}^d \to \mathbb{R}$ es lineal, son convexos.

c) La intersección de conjuntos convexos es convexa.

d) El interior y el cierre de conjuntos convexos es convexo.

e) Si $K$ es convexo y $K^{\mathrm{o}} \neq \emptyset$, entonces $\overline{K^{\mathrm{o}}} = \overline{K}$ y $\overline{K}^{\mathrm{o}} = K^{\mathrm{o}}$.

*Demostración.*

a) Es evidente por la definición de subespacio vectorial.

b) Lo hacemos para $K = \{x \in V : T(x) < \alpha\}$. Para el resto de desigualdades es análogo. Sean $x_1, x_2 \in K$ y $\lambda \in [0,1]$. Entonces, $T(x_1), T(x_2) < \alpha$. Por la linealidad de $T$ se tiene que $T((1-\lambda)x_1 + \lambda x_2) = (1-\lambda)T(x_1) + \lambda T(x_2) < (1-\lambda)\alpha + \lambda\alpha = \alpha$, luego $(1-\lambda)x_1 + \lambda x_2 \in K$.

c) Sea $\{K_i\}_{i\in I}$ una familia de conjuntos convexos. Si su intersección es vacía, no hay nada que probar. En caso contrario, tomamos $x_1, x_2 \in \bigcap_{i\in I} K_i$ y $\lambda \in [0,1]$. Se tiene que $x_1, x_2 \in K_i$ para todo $i \in I$, luego $(1-\lambda)x_1 + \lambda x_2 \in K_i$ para todo $i \in I$ y por tanto $(1-\lambda)x_1 + \lambda x_2 \in \bigcap_{i\in I} K_i$, concluyendo que la intersección es convexa.

d) Sea $K$ convexo y $x, y \in \overline{K}$. Existen por tanto sucesiones $\{x_n\} \to x, \{y_n\} \to y$, con $x_n, y_n \in K$, para todo $n \in \mathbb{N}$. Entonces, $[x_n, y_n] \subset K$ para todo $n \in \mathbb{N}$, y por tanto, $[x, y] \subset \overline{K}$, luego $\overline{K}$ es convexo.

Supongamos $K^{\circ} \neq \emptyset$ y sean $x, y \in K^{\circ}$. Existe por tanto $\varepsilon > 0$ tal que $B(x,\varepsilon) \subset K$ y $B(y,\varepsilon) \subset K$. Tomamos $z, w \in B(0,\varepsilon)$. Es claro que $z + x \in B(x,\varepsilon)$ y $w + y \in B(y,\varepsilon)$. Sea $\lambda \in [0,1]$. Se tiene que

$$\begin{aligned} K \ni (1-\lambda)(z+x) + \lambda(w+y) &= (1-\lambda)z + \lambda w + (1-\lambda)x + \lambda y \\ &\implies ((1-\lambda)x + \lambda y) + (1-\lambda)B(0,\varepsilon) + \lambda B(0,\varepsilon) \subset K \\ &\implies B((1-\lambda)x + \lambda y, \varepsilon) = B(0,\varepsilon) + (1-\lambda)x + \lambda y \subset K. \end{aligned}$$

Por tanto, $(1-\lambda)x + \lambda y \in K^{\circ}$ y $K^{\circ}$ es convexo.

e) Es claro que $\overline{C^{\circ}} \subset \overline{C}$. Para la inclusión recíproca, tomamos $x \in \overline{C}$ y $U$ un entorno arbitrario de $x$. Entonces $U \cap C \neq \emptyset$. Tomamos $y \in U \cap C$, y $z \in C^{\circ}$. Se tiene que $(1-\lambda)z + \lambda y \in C^{\circ}$ para todo $0 \leq \lambda < 1$, y en consecuencia $U \cap C^{\circ} \neq \emptyset$, luego $x \int \overline{C^{\circ}}$.

Por otra parte, es claro que $C^{\circ} \subset \overline{C}^{\circ}$. Para la inclusión recíproca, tomamos $x \in \overline{C}^{\circ}$. Entonces existe $\varepsilon > 0$ tal que $B(x,\varepsilon) \subset \overline{C}$. Sea $y \int C^{\circ}$. Entonces, existe $\delta > 0$ tal que $y + (1+\delta)(x-y) \in B(x,\varepsilon) \subset \overline{C}$. Como $y \in \int C$, se tiene que $y + \lambda(1+\delta)(x-y) \in C^{\circ}$, para $0 <\leq \lambda < 1$. En particular, $x = y + (1/1+\delta)(1+\delta)(x-y) \in C^{\circ}$.

□

Una caracterización muy conocida de los conjuntos convexos, y que es la extensión natural de la definición de convexidad, permite afirmar que los conjuntos convexos son cerrados respecto a un tipo de combinaciones lineales que definimos a continuación.

**Definición 1.1.3.** Dados $x_1, \ldots, x_k \in \mathbb{R}^d$, una combinación convexa de $x_1, \ldots, x_k$ es una combinación lineal $x$ de $x_1, \ldots, x_k$ donde los coeficientes suman 1, esto es, $x = \sum_{i=1}^k \lambda_i x_i$, con $\sum_{i=1}^k \lambda_i = 1$. A los coeficientes $\lambda_i$ se les denomina coordenadas baricéntricas de $x$ respecto a $x_1, \ldots, x_k$.

**Proposición 1.1.1.** *Un subconjunto $K \subset \mathbb{R}^d$ es convexo si y solo si toda combinación convexa de puntos de $K$ pertenece a $K$.*
*Demostración.*

$\Leftarrow$) Es un caso particular, tomando dos puntos.

$\Rightarrow$) $\sum_{i=1}^k \lambda_i x_i = (1-\lambda_k)\left(\sum_{i=1}^{k-1} \frac{\lambda_i}{1-\lambda_k} x_i\right) + \lambda_k x_k$ y aplicamos inducción (notemos que $\sum_{i=1}^{k-1} \frac{\lambda_i}{1-\lambda_k} = (1-\lambda_k)/(1-\lambda_k) = 1$).

□

### 1.1.2. Hiperplanos soporte

Nuestro objetivo ahora es probar una caracterización aún más fuerte para los conjuntos convexos, a través de hiperplanos. Es conocido que los conjuntos convexos verifican que los hiperplanos que tocan el conjunto "tangencialmente" dejan el conjunto completo "a un lado" del hiperplano. Vamos a formalizar este concepto, y a probar que esto caracteriza a los conjuntos convexos con interior no vacío.

**Definición 1.1.4.** Sea $T\colon \mathbb{R}^d \to \mathbb{R}$ una aplicación lineal, $\alpha \in \mathbb{R}$ y $P = \{x \in \mathbb{R}^d\colon T(x) = \alpha\}$ un hiperplano. Asociados a $P$, definimos los semiespacios $P^+ = \{x \in \mathbb{R}^d\colon T(x) \geq \alpha\}$ y $P^- = \{x \in \mathbb{R}^d\colon T(x) \leq \alpha\}$.

Diremos que $P$ es un *hiperplano soporte* para el conjunto $K \subset \mathbb{R}^d$ si $P \cap \overline{K} \neq \emptyset$ y $K \subset P^+$ o $K \subset P^-$. Al semiespacio que lo contiene, de entre $P^+$ y $P^-$, se denomina *semiespacio soporte*.

Notemos que la definición de hiperplano soporte es un concepto topológico que modela, sin nociones de diferenciabilidad, el concepto de "tangencialidad". Cuando $K$ tiene interior no vacío, dicho interior está contenido en uno de los semiespacios, sin llegar a cortar al hiperplano, pues las bolas de $\mathbb{R}^d$ no pueden estar contenidas en hiperplanos. En tales casos, los hiperplanos soporte tocan a $K$ únicamente en la frontera. Esta idea de tangencialidad es la que describen estos hiperplanos.

Pasamos a enunciar el teorema con la caracterización que habíamos anticipado. Antes necesitaremos recordar un resultado sobre las distancias a conjuntos cerrados.

**Proposición 1.1.2.** *Sea $K \subset \mathbb{R}^d$ un subconjunto cerrado no vacío. Entonces, para cada $x \in \mathbb{R}^d$ existe $x_0 \in K$ tal que $d(x, x_0) = d(x, K)$, donde la distancia a conjunto viene definida por*

$$d(x, K) = \text{ínf}\{\|x - y\|\colon y \in K\}.$$

*Es decir, en los conjuntos cerrados no vacíos hay puntos que materializan la distancia a dicho conjunto.*
*Demostración.* Sea $x \in \mathbb{R}^d$. Como $K$ es cerrado, podemos tomar $R > 0$ tal que $K \cap \overline{B}(x, R)$ es compacto y no vacío. Podemos considerar la función distancia a $x$ sobre dicho conjunto, $d_x\colon K \cap \overline{B}(x, R) \to \mathbb{R}_0^+$, dada por $d_x(y) = d(x, y) = \|x - y\|$. $d_x$ es continua, por serlo la aplicación norma y las traslaciones, y está definida sobre un compacto, luego alcanza un mínimo en $x_0 \in K \cap \overline{B}(x, R)$.

Si ahora tomamos $y \in K \cap \overline{B}(x, R)$, se tiene que $d(x, y) = d_x(y) \geq d_x(x_0) = d(x, x_0)$. Por otro lado, si tomamos $y \in K \setminus \overline{B}(x, R)$, se tiene que $d(x, y) > r \geq d(x, x_0)$. Por tanto, $d(x, y) \geq d(x, x_0)$ para todo $y \in K$, luego $d(x, K) \geq d(x, x_0)$. La otra desigualdad es clara, pues $x_0 \in K$. Por tanto, $x_0$ es el punto buscado. □

**Teorema 1.1.3** (Teorema del hiperplano soporte)**.**

a) *Si $K \subset \mathbb{R}^d$ es convexo y cerrado, para cada $x_0 \in \operatorname{Fr} K$ existe un hiperplano soporte $P$ de $K$ tal que $x_0 \in P$.*

b) *Todo conjunto convexo cerrado propio de $\mathbb{R}^d$ es la intersección de todos sus semiespacios soporte.*

c) *Sea $K \subset \mathbb{R}^d$ un conjunto cerrado con interior no vacío. Entonces, $K$ es convexo si y solo si para todo $x \in \operatorname{Fr} K$ existe un hiperplano soporte $P$ de $K$ con $x \in P$.*

*Demostración.*

a) Si $K = \emptyset$ o $K = \mathbb{R}^d$, la frontera es vacía y no hay nada que probar. En otro caso, podemos tomar $x_0 \in \operatorname{Fr} K$ y una sucesión de puntos $\{y_n\}$ en $\mathbb{R}^d \setminus K$ de forma que $\{y_n\} \to x_0$. Además, como $K$ es cerrado, existen puntos en $K$ en los que se materializa la distancia de $K$ a cualquier punto. En particular, para cada $y_n$, exist $x_n \in K$ tal que $\|y_n - x_n\| = d(y_n, K)$. Consideramos

la sucesión $\{x_n\} \subset K$ con tales puntos, y la sucesión $\{e_n\} = \{(y_n - x_n)/(\|y_n - x_n\|)\}$. $\{e_n\}$ está bien definida, pues $x_n \neq y_n$ para todo $n \in \mathbb{N}$, y $|e_n| = 1$ para todo $n \in \mathbb{N}$. Además, $\|x_n - x_0\| \leq \|x_n - y_n\| + \|y_n - x_0\| \to 0$, luego $\{x_n\} \to x_0$.

Observemos que, dado $x \in K$, el segmento $[x, x_n] \subset K$, para todo $n \in \mathbb{N}$. Como $x_n$ minimiza la distancia a $y_n$ en $K$, la función $\phi\colon [0,1] \to \mathbb{R}$ dada por $\phi(\lambda) = \|y_n - (\lambda x + (1-\lambda)x_n)\|^2$ alcanza un mínimo absoluto en 0, luego existe $\varepsilon > 0$ tal que $\phi$ es creciente en $]0, \varepsilon[$, y por tanto $\phi'(0) \geq 0$, esto es,

$$2\langle -(x - x_n), y_n - x_n\rangle \geq 0 \iff \langle x - x_n, y_n - x_n\rangle \leq 0 \iff \langle x - x_n, e_n\rangle \leq 0 \quad \forall x \in K.$$

Como $\{e_n\}$ está acotada, por el teorema de Bolzano-Weierstrass existe una parcial convergente, $\{e_{\sigma(n)}\} \to e$. Si consideramos $\{x_{\sigma(n)}\} \to x_0$, tomando límites y utilizando la continuidad del producto escalar, se tiene que $\langle x - x_0, e\rangle \leq 0$, para todo $x \in K$.

Por tanto, $K \subset \{x \in \mathbb{R}^d\colon \langle x - x_0, e\rangle \leq 0\}$ y $x_0 \in K \cap \{x \in \mathbb{R}^d\colon \langle x - x_0, e\rangle = 0\}$, luego el hiperplano perpendicular a $e$ que pasa por $x_0$ es un hiperplano soporte que contiene a $x_0$.

b) Supongamos $K$ convexo, cerrado y propio (es decir, $K \neq \mathbb{R}^d$ y $K \neq \emptyset$). Entonces, $\operatorname{Fr} K \neq \emptyset$ y por a) existen hiperplanos soporte que contienen a $K$. Llamamos $K'$ a la intersección de todos los semiespacios soporte asociados. Es claro que $K'$ es cerrado y convexo, y $K \subset K'$. Supongamos que existe $x' \in K' \setminus K$. Como $K$ es cerrado, existe $x_0 \in K$ que materializa la distancia de $x'$ a $K$. Razonando como en a), se obtiene que

$$K \subset S = \{x \in \mathbb{R}^d\colon \langle x' - x_0, x - x_0\rangle \leq 0\},$$

luego $S$ es un hiperplano soporte de $K$. Por otra parte, como $K'$ es intersección de hiperplanos soporte, se tiene que $K' \subset S$. En particular, $x' \in S$, pero entonces

$$0 < \|x' - x_0\|^2 = \langle x' - x_0, x' - x_0\rangle \leq 0,$$

llegando a una contradicción.

c)

$\Rightarrow$) Es consecuencia de a).

$\Leftarrow$) Sea $K$ cerrado con $K^{\circ} \neq \emptyset$ y supongamos que $K$ no es convexo. En particular, $K \neq \emptyset$ y $K \neq \mathbb{R}^d$, luego $\operatorname{Fr} K \neq \emptyset$. Como $K$ es no convexo, existen $x_1, x_2 \in K$ y $x \in [x_1, x_2]$ con $x \notin K$. Tomamos $x' \in K^{\circ}$ y consideramos el segmento $[x, x']$. Como $x' \in K^{\circ}$ y $x \in \mathbb{R} \setminus K = (R \setminus K)^{\circ}$, se tiene que $[x, x'] \cap \operatorname{Fr} K \neq \emptyset$, luego podemos tomar $x_0 \in \operatorname{Fr} K \cap [x, x']$. Veamos que $x_0$ no admite un hiperplano soporte.

En efecto, supongamos que existe tal hiperplano $P$, y llamamos $H$ al correspondiente semiespacio soporte. Como $K \subset H$, $K^{\circ} \subset H^{\circ}$, y además $H^{\circ} \cap = \emptyset$, luego como $x' \in K^{\circ}$, $x' \notin P$. Por tanto, $[x', x] \not\subset P$, luego su intersección es, a lo sumo, un punto, y necesariamente a de ser $[x, x'] \cap P = \{x_0\}$. Por otra parte, $x \notin H$, pues de estarlo, solo podría estar en $H^{\circ}$, y al ser convexo, implicaría también $x_0 \in H^{\circ}$, lo que no es posible.

Por tanto, o bien $x_1$ o bien $x_2$ no están en $H$, pues en caso contrario $x \in [x_1, x_2]$ debería estarlo también, pero esto contradice que $H$ sea un hiperplano soporte, pues $x_1, x_2 \in K$.

□

Para concluir, hay que destacar que la condición $K^o \neq \emptyset$ no se puede eliminar en el apartado *c)* del teorema. Por ejemplo, la gráfica de la función exponencial en $\mathbb{R}^2$ es no convexa, cerrada, su interior es vacío, y admite rectas soporte en cada punto (las tangentes en dichos puntos). En general, si $K$ es convexo y cerrado con interior no vacío, su frontera tiene interior vacío, no es necesariamente convexa y tiene en cada punto los mismos hiperplanos soporte que $K$.

### 1.1.3. Proyecciones convexas

Hemos visto en la proposición 1.1.2 que los conjuntos cerrados permiten, para cada punto $x \in \mathbb{R}^d$, expresar la distancia al conjunto como $d(x, x_0)$, donde $x_0$ pertenece al conjunto. Cuando cada punto admite un único $x_0$, podemos definir una aplicación en $\mathbb{R}^d$ que envía cada punto al único punto más cercano dentro del conjunto. Esto es lo que se conoce como una *proyección*.

En general, no tenemos proyecciones definidas sobre cualquier conjunto cerrado, pues puede haber varios puntos donde se materialice la distancia (consideremos por ejemplo como conjunto una circunferencia en el plano, donde la distancia al centro se materializa en todos los puntos). Sí sabemos que, en espacios de Hilbert, la proyección sobre subespacios cerrados está bien definida, gracias al teorema de la proyección ortogonal, que añade además determinadas condiciones de ortogonalidad. Vamos a ver que estos no son los únicos subespacios que admiten proyecciones, sino que estas proyecciones están bien definidas en cualquier convexo cerrado.

**Teorema 1.1.4** (Teorema de la proyección convexa)**.** *Si $K \subset \mathbb{R}^d$ es no vacío, cerrado y convexo, entonces, para cada $x \in \mathbb{R}^d$ existe un único punto $x_0 \in K$ tal que $d(x, K) = d(x, x_0)$. Al punto $x_0$ se le denomina la* proyección *de $x$ sobre $K$ y se suele notar $P_K(x)$, y la aplicación $P_K \colon \mathbb{R}^d \to K$ que realiza la asignación $x \mapsto P_K(x)$ se denomina la proyección sobre $K$.*

*Además, para cada $x \in \mathbb{R}^d \setminus K$, el semiespacio $\{y \in \mathbb{R}^d \colon \langle x - P_K(x), y - P_k(x)\rangle \leq 0\}$ es un semiespacio soporte de $K$ en $P_K(x)$.*

*Demostración.* La existencia nos la da la proposición 1.1.2. Veamos la unicidad. Sea $x \in \mathbb{R}^d$ y supongamos que $x_1, x_2 \in K$ verifican $d(x, x_1) = d(x, K) = d(x, x_2)$. Tomamos $x_0$ como el punto medio del segmento $[x_1, x_2]$. Se tiene que $x_0 \in K$ por ser $K$ convexo. Observemos que

$$\langle x_1 - x_2, x - x_0\rangle = \langle x_1 - x_2, x - \frac{1}{2}(x_1 + x_2)\rangle = \frac{1}{2}\langle x_1 - x_2, 2x - x_1 - x_2\rangle.$$

Si sustituimos $x_1 - x_2 = (x - x_2) - (x - x_1)$ y $2x - x_1 - x_2 = (x - x_2) + (x - x_1)$, obtenemos

$$\begin{aligned}\langle x_1 - x_2, x - x_0\rangle &= \frac{1}{2}\langle (x - x_2) - (x - x_1), (x - x_2) + (x - x_!)\rangle \\ &= \frac{1}{2}(\|x - x_2\|^2 - \|x - x_1\|^2) \\ &= \frac{1}{2}(d(x, K)^2 - d(x, K)^2) = 0.\end{aligned}$$

Por tanto, los vectores $x_1 - x_2$ y $x - x_0$ son ortogonales, y en consecuencia también lo son $x - x_0$ y $x_0 - x_2 = (x_1 - x_2)/2$. Aplicando el teorema de pitágoras, obtenemos

$$d(x, K)^2 = \|x - x_2\|^2 = \|x - x_0\|^2 + \|x_0 - x_2\|^2 \geq \|x - x_0\|^2 \geq d(x, K)^2.$$

Por tanto, se da la igualdad en las desigualdades anteriores, obteniendo en particular que $\|x_0 - x_2\|^2 = 0$, luego $x_0 = x_2$. Como $x_0$ era el punto medio de $[x_1, x_2]$, se concluye que $x_1 = x_2$, probando la unicidad.

Finalmente, sea $x \in \mathbb{R}^d \setminus K$ y supongamos que existe $y \in K$ con $\langle x - P_K(x), y - P_K(x)\rangle > 0$. Por ser $K$ convexo, el segmento $[y, P_K(x)]$ está contenido en $K$, luego los puntos de la forma $y_t = P_K(x) + t(y - P_K(x)) \in K$, para todo $t \in [0,1]$. Definimos la aplicación $f\colon [0,1] \to \mathbb{R}$, por

$$f(t) = \|y_t - x\|^2 = \|P_K(x) - x + t(y - P_K(x))\|^2 = \|p_K(x) - x\|^2 + 2t\langle P_K(x) - x, y - P_K(x)\rangle + t^2\|y - P_K(x)\|^2.$$

$f$ es un polinomio en $t$, luego es diferenciable, y

$$f'(0) = 2\langle P_K(x) - x, y - P_K(x)\rangle = -2\langle x - P_K(x), y - P_K(x)\rangle < 0.$$

Por tanto, $f$ es estrictamente decreciente en un entorno de 0, esto es, existe $\varepsilon > 0$ tal que $\|y_t - x\|^2 < \|y_0 - x\|^2 = \|P_K(x) - x\|^2$ para $0 < t < \varepsilon$, llegando a una contradicción, pues en $P_K(x)$ se minimiza la distancia a $x$ en $K$, y los $y_t$ pertenecen a $K$. □

Para concluir esta sección, es interesante destacar que, además de que todos los conjuntos convexos cerrados admiten una proyección, esta propiedad los caracteriza. Es decir, todo conjunto cerrado de $\mathbb{R}^d$ que, para cada punto $x$ en $\mathbb{R}^d$ admita un único punto donde se materialice la distancia al conjunto, es convexo. Este resultado, que no vamos a utilizar, se conoce como teorema de Bunt-Motzkin [10].

### 1.1.4. Conos

En esta sección presentaremos un tipo especial de conjuntos convexos, con unas propiedades muy interesantes, y de gran importancia en optimización.

**Definición 1.1.5** (Conos)**.** Un subconjunto $C \subset \mathbb{R}^d$ se dice que es *cónico* si para cada $x \in C$ y cada $\alpha \in \mathbb{R}_0^+$, se tiene que $\alpha x \in C$.

Un subconjunto $C \subset \mathbb{R}^d$ se dice que es un *cono* si es cónico y convexo. Esto es equivalente a decir que para cada $x, y \in C$ y cualesquiera $\alpha, \beta \in \mathbb{R}_0^+$, se tiene que $\alpha x + \beta y \in C$.

Una *combinación cónica* de $x_1, \ldots, x_k \in \mathbb{R}^d$ es una combinación lineal de la forma $\alpha_1 x_1 + \cdots + \alpha_k x_k$, con $\alpha_1, \ldots, \alpha_k \in \mathbb{R}_0^+$. Es inmediato ver que un conjunto es un cono si y solo si es cerrado para las combinaciones cónicas.

En algunos textos a los conjuntos cónicos se les denomina inicialmente conos, y a aquellos convexos se les denomina conos convexos. En este trabajo el término cono se reservará únicamente para estos últimos. Observemos también que con esta definición todos los conjuntos cónicos y conos contienen al 0. Veamos algunos ejemplos de conjuntos cónicos y conos.

Ejemplo 1.1.5:

a) Un conjunto finito o numerable de rectas o semirrectas en $\mathbb{R}^2$ que pasan por 0 (siendo este el origen en el caso de las semirrectas) es un conjunto cónico, pero no es un cono.

b) Los subespacios vectoriales son conos.

c) El conjunto de los números reales no negativos, $\mathbb{R}_0^+$, es un cono.

d) Los cuadrantes u octantes del plano o el espacio, incluyendo al 0 (sin ser necesariamente cerrados) son conos. Más en general, el conjunto

$$(\mathbb{R}^d)_0^+ = \{(x_1, \ldots, x_d) \in \mathbb{R}^d\colon x_i \geq 0, i = 1, \ldots, d\}$$

es un cono.

e) El conjunto de los (coeficientes de) polinomios no negativos de grado par,

$$(P_{2d})_0^+ = \{(a_0, a_1, \ldots, a_{2d}) \in \mathbb{R}^{2d+1} \colon a_0 + a_1 x + a_2 x^2 + \cdots + a_{2d} x^{2d} \geq 0 \quad \forall x \in \mathbb{R}\}$$

es un cono.

Dentro de los conos, podemos destacar una familia especial, cuyos elementos se denominan conos propios.

**Definición 1.1.6.** Sea $C \subset \mathbb{R}^d$ un cono.

- $C$ es *sólido* si tiene interior no vacío.
- $C$ es *puntiaguado* si $C \cap (-C) = \{0\}$.
- $C$ es *propio* si es cerrado, sólido y puntiagudo.

Los conjuntos $\mathbb{R}_0^+$, $(\mathbb{R}^d)_0^+$ y $(P_{2d})_0^+$ del ejemplo 1.1.5 son conos propios. Los conos propios permiten definir una relación de orden sobre el espacio vectorial donde está definido el cono, de forma que con dicha relación de orden, el cono se puede entender como un conjunto de números "positivos" sobre dicho espacio, generalizando así a los números reales positivos. Para ello, fijamos un cono $C \subset \mathbb{R}^d$ y definimos la relación $\preceq$ de forma que para $x, y \in \mathbb{R}^d$, $x \preceq y \iff y - x \in C$. Veamos que $\preceq$ es una relación de orden.

- Es reflexiva: $x - x = 0 \in C$, luego $x \preceq x$.
- Es antisimétrica: si $x \preceq y$ e $y \preceq x$, entonces $y - x \in C$, $x - y \in C$, luego $y - x \in C \cap (-C) = \{0\}$ y por tanto $x = y$.
- Es transitiva: si $x \preceq y$ e $y \preceq z$, entonces $z - y, y - x \in C$, luego $z - x = (z - y) + (y - x) \in C$ y por tanto $x \preceq z$.

Sin embargo, este orden no es en general un orden total. Podemos definir también un orden estricto asociado a $C$, dado por $x \prec y \iff y - x \in C^{\mathrm{o}}$. De forma análoga se puede comprobar que $x \not\prec x$, que $x \prec y \implies y \not\prec x$, y que de nuevo es transitivo. Ambos órdenes respetan además la suma y el producto por escalares no negativos en el espacio vectorial, es decir,

- $x \preceq y, z \preceq w \implies x + z \preceq y + w$.
- $x \prec y, z \preceq w \implies x + z \prec y + w$.
- $x \preceq y, \alpha \in \mathbb{R}_0^+ \implies \alpha x \preceq \alpha y$.
- $x \prec y, \alpha \in \mathbb{R}^+ \implies \alpha x \prec \alpha y$.

Además, el orden también respeta la convergencia:

- Si $\{x_n\}, \{y_n\}$ son sucesiones en $\mathbb{R}^d$ con $\{x_n\} \to x$ y $\{y_n\} \to y$, y $x_n \preceq y_n$ para todo $n \in \mathbb{N}$ o $x_n \prec y_n$ para todo $n \in \mathbb{N}$, entonces $x_n \preceq y_n$.

*Demostración.* Se tiene que $x_n - y_n \in C$ (resp. $C^{\mathrm{o}}$) para todo $n \in \mathbb{N}$ y $C$ es cerrado, luego $x - y \in \overline{C} = C$ (resp. $x - y \in \overline{C^{\mathrm{o}}} = \overline{C} = C$). □

Para concluir, veamos cómo se manifiestan estos órdenes en los ejemplos de 1.1.5.

Ejemplo 1.1.6:

- El orden inducido por $\mathbb{R}_0^+$ sobre $\mathbb{R}$ es el orden usual de los números reales.

- El orden inducido por $(\mathbb{R}^d)_0^+$ sobre $\mathbb{R}^d$ es el orden producto (es decir, $x \preceq y \iff x_i \leq y_i \quad \forall i = 1, \ldots, d$). En este caso observamos que el orden no es total.
- El cono de los polinomios no negativos de grado par induce un orden sobre los vectores de coeficientes que es equivalente al orden como funciones de los polinomios asociados.

## 1.2. Funciones convexas

### 1.2.1. Definición y propiedades

En esta sección recordaremos el concepto de funciones convexas, sus propiedades más conocidas, presentando algunas funciones convexas que serán de utilidad más adelante, junto a distintos métodos para reconocerlas.

**Definición 1.2.1.** Sea $K \subset \mathbb{R}^d$ un subconjunto convexo.

Una función $f\colon K \to \mathbb{R}^d$ diremos que es *convexa* si para todos $x, y \in K$ y cada $\lambda \in [0, 1]$, se tiene

$$f((1-\lambda)x + \lambda y) \leq (1-\lambda)f(x) + \lambda f(y).$$

Diremos que $f$ es *estrictamente convexa* si la desigualdad anterior es estricta, es decir, si para cuaesquiera $x, y \in K$ con $x \neq y$ y cada $\lambda \in ]0, 1[$ se tiene

$$f((1-\lambda)x + \lambda y) < (1-\lambda)f(x) + \lambda f(y).$$

Cuando las desigualdades en las expresiones anteriores se den en dirección contraria, diremos que $f$ es *cóncava* o *estrictamente cóncava*. Es claro que $f$ es cóncava (resp. estrictamente cóncava) si y solo si $-f$ es convexa (resp. estrictamente convexa), luego una teoría análoga a la de las funciones convexas puede realizarse para las funciones cóncavas. Por ello, nos centraremos únicamente en funciones convexas.

Notemos que todas las definiciones anteriores son correctas, pues el dominio es convexo, y por tanto tiene sentido evaluar $f$ a lo largo del segmento $[x, y]$. También hay que destacar la interpretación geométrica de las definiciones, cuando nos restringimos a una variable, la cual nos dice que la gráfica de la función $f$ entre $x$ e $y$ está siempre por debajo del segmento que une los puntos $(x, f(x))$ e $(y, f(y))$, estando estrictamente por debajo, salvo en los extremos, cuando la función es estrictamente convexa.

Veamos en primer lugar distintas formas de caracterizar a las funciones convexas.

**Proposición 1.2.1.** *Sea $K \subset \mathbb{R}^d$ y $f\colon K \to \mathbb{R}$. Entonces, son equivalentes:*

a) *$K$ es convexo y $f$ es convexa.*

b) *El* epigrafo *de $f$ es un conjunto convexo, donde el epigrafo asociado a una función $f\colon K \to \mathbb{R}$ viene dado por*

$$\mathrm{Epi}(f) = \{(x, y) \in K \times \mathbb{R}\colon y \geq f(x)\}.$$

c) *$K$ es convexo, y para cualesquiera $x_1, x_2 \in K$, la función $\varphi\colon [0, 1] \to \mathbb{R}$ dada por $\varphi(t) = f((1-t)x_1 + tx_2)$ es convexa (análogamente se tiene la desigualdad estricta si suponemos convexidad estricta)*

*Demostración.*

*a)* $\implies$ *b)*: Si $(x_1, y_1), (x_2, y_2) \in \mathrm{Epi}(f)$, entonces $f(x_1) \leq y_1$ y $f(x_2) \leq y_2$. Sea $\lambda \in [0,1]$. Por la convexidad de $f$, $f((1-\lambda)x_1 + \lambda x_2) \leq (1-\lambda)f(x_1) + \lambda f(x_2) \leq (1-\lambda)y_1 + \lambda y_2$, y por tanto, $(1-\lambda)(x_1, y_1) + \lambda(x_2, y_2) \in \mathrm{Epi}(f)$.

*b)* $\implies$ *a)*: La aplicación $\pi\colon \mathbb{R}^{d+1} \to \mathbb{R}^d$ dada por $\pi(x,t) = x$ es lineal, y $\pi(\mathrm{Epi}(f)) = K$. Es claro que las aplicaciones lineales conservan los conjuntos convexos, luego $K$ es convexo. Dados $x_1, x_2 \in K$, la convexidad de $f$ se deduce considerando los puntos $(x_1, f(x_1)), (x_2, f(x_2)) \in \mathrm{Epi}(f)$ y usando la convexidad de este.

*a)* $\implies$ *c)*: Fijamos $x_1, x_2 \in K$, y sean $\lambda, t, s \in [0,1]$. Entonces,

$$\begin{aligned}
\varphi((1-\lambda)t + \lambda s) &= f([1-(1-\lambda)t - \lambda s]x_1 + [(1-\lambda)t + \lambda s]x_2) \\
&= f([(1-(1-\lambda)t)x_1 + (1-\lambda)tx_2] + [\lambda s x_2 - \lambda s x_1]) \\
&= f((1-(1-\lambda)t - \lambda)x_1 + (1-\lambda)tx_2] + [\lambda s x_2 + \lambda(1-s)x_1]) \\
&= f((1-\lambda)((1-t)x_1 + tx_2) + \lambda((1-s)x_1 + sx_2)) \\
&\leq (1-\lambda)\varphi(t) + \lambda\varphi(s).
\end{aligned}$$

*c)* $\implies$ *a)*: Para $x_1, x_2 \in K$ y $\lambda \in [0,1]$, se tiene

$$\begin{aligned}
f((1-\lambda)x_1 + \lambda x_2) = \varphi(\lambda) &= \varphi((1-\lambda)\cdot 0 + \lambda \cdot 1) \\
&\leq (1-\lambda)\varphi(0) + \lambda\varphi(1) = (1-\lambda)f(x_1) + \lambda f(x_2).
\end{aligned}$$

□

Los siguientes resultados bien conocidos sobre funciones convexas las relacionan con dos campos en los que son de gran interés: la optimización y la diferenciabilidad.

**Proposición 1.2.2.**

a) *Todo mínimo local de una función convexa es un mínimo global.*

b) *Toda función estrictamente convexa tiene a lo sumo un mínimo local, que también será global.*

c) *Toda función convexa en un conjunto convexo y abierto es localmente lipschitziana. En particular, es continua.*

d) *Sea $\Omega \subset \mathbb{R}^d$ abierto y convexo, y sea $f\colon \Omega \to \mathbb{R}$ una función de clase $\mathcal{C}^1(\Omega)$. Entonces, $f$ es convexa si y solo si para cualesquiera $x, x_0 \in \Omega$, se tiene*

$$f(x) \geq f(x_0) + \langle \nabla f(x_0), x - x_0 \rangle \quad \forall x, x_0 \in \Omega.$$

*Esto último se interpreta geométricamente como que el grafo de $f$ permanece por encima del plano tangente a $f$ en $x_0$. Si $f$ además es estrictamente convexa, la desigualdad es estricta siempre que $x \neq x_0$.*

e) *Sea $\Omega \subset \mathbb{R}^d$ abierto y convexo, y sea $f\colon \Omega \to \mathbb{R}$ una función de clase $\mathcal{C}^2(\Omega)$. Entonces, $f$ es convexa si y solo si su matriz hessiana es semidefinida positiva en todo punto de $\Omega$.*

f) *Sea $\Omega \subset \mathbb{R}^d$ abierto y convexo, y sea $f\colon \Omega \to \mathbb{R}$ una función de clase $\mathcal{C}^2(\Omega)$. Entonces, $f$ es estrictamente convexa si su matriz hessiana es definida positiva en todo punto de $\Omega$. El recíproco no es cierto en general.*

Observemos que las funciones convexas aportan propiedades sobre la globalidad y unicidad de los mínimos, sin afirmar nada de su existencia. Para la existencia será necesario recurrir a otros argumentos, como la continuidad y la compacidad.

Para concluir, vamos a ver algunos ejemplos de funciones convexas que nos serán de utilidad más adelante, junto con las operaciones que preservan la convexidad.

Ejemplo 1.2.3:

a) Las aplicaciones afines en $\mathbb{R}^d$ son cóncavas y convexas. De hecho, estas son las únicas aplicaciones cóncavas y convexas simultáneamente.

b) La función exponencial, $x \mapsto e^{\alpha x}$, es estrictamente convexa en $\mathbb{R}$, para todo $\alpha \in \mathbb{R}$.

c) La función logaritmo es estrictamente cóncava en $\mathbb{R}^+$.

d) Las normas en $\mathbb{R}^d$ son convexas.

e) La función *logaritmo-suma-exponencial*, $f\colon \mathbb{R}^d \to \mathbb{R}$, dada por $f(x_1, \ldots, x_d) = \log(\sum_{i=1}^d e^{x_i})$, es convexa.

f) Las combinaciones lineales con coeficientes no negativos de funciones convexas son convexas.

g) Si $K \subset \mathbb{R}^d$ es convexo, $f\colon K \to \mathbb{R}$ es convexa, $A \subset \mathbb{R}^n$ y $h\colon A \to \mathbb{R}^d$ es afín, con $h(A) \subset K$, entonces $f \circ h$ es convexa.

h) Si $K \subset \mathbb{R}^d$ es convexo y $f_1, \ldots, f_n\colon K \to \mathbb{R}$ son convexas, entonces $f\colon K \to \mathbb{R}$, dada por $f(x) = \text{máx}\{f_1(x), \ldots, f_n(x)\}$ es convexa.

La convexidad de las normas es consecuencia de la desigualdad triangular. La convexidad de la función logaritmo-suma-exponencial puede obtenerse mediante el cálculo de la matriz Hessiana. Esta función será de gran importancia en algunos de los algoritmos que veremos más adelante. El resto de ejemplos son propiedades conocidas de las funciones convexas.

### 1.2.2. La desigualdad de Jensen

A continuación probaremos la conocida como desigualdad de Jensen. La versión más particular de esta desigualdad es una generalización de la desigualdad dada en la definición de funciones convexas, y puede probarse fácilmente por inducción. La versión que vamos a demostrar es algo más general, siendo así válida para medidas de probabilidad. Para ello utilizaremos herramientas de la teoría de la medida.

En primer lugar, veamos una propiedad que verifican las funciones convexas de variable real.

**Lema 1.2.4** (Lema de las tres secantes)**.** *Sean $-\infty \leq a < b \leq \infty$ y $\varphi\colon ]a, b[ \to \mathbb{R}$ una función convexa. Entonces,*

$$\frac{\varphi(t) - \varphi(s)}{t - s} \leq \frac{\varphi(u) - \varphi(s)}{u - s} \leq \frac{\varphi(u) - \varphi(t)}{u - t}, \text{ para cualesquiera } a < s < t < u < b.$$

*Demostración.* Dados $t_1, t_2 \in ]a, b[$, y $t_0 \in [t_1, t_2]$, podemos tomar $\lambda = (t_0 - t_1)/(t_2 - t_1) \in [0, 1]$, y $1 - \lambda = (t_2 - t_0)/(t_2 - t_1)$, verificándose que $t_0 = (1 - \lambda)t_1 + \lambda t_2$ lo que nos permite expresar la convexidad de $\varphi$ mediante la expresión

$$\varphi(t_0) \leq \frac{t_2 - t_0}{t_2 - t_1}\varphi(t_1) + \frac{t_0 - t_1}{t_2 - t_1}\varphi(t_2). \tag{1.1}$$

Sean $a < s < t < u < b$. Si aplicamos la ecuación 1.1 con $s, t$ y $u$, obtenemos

$$\varphi(t) \leq \frac{u-t}{u-s}\varphi(s) + \frac{t-s}{u-s}\varphi(u). \tag{1.2}$$

Restando $\varphi(s)$ y dividiendo por $t-s$, se tiene

$$\begin{aligned}\frac{\varphi(t)-\varphi(s)}{t-s} &\leq \frac{1}{t-s}\left(\frac{u-t}{u-s}\varphi(s) + \frac{t-s}{u-s}\varphi(u) - \varphi(s)\right)\\ &= \frac{1}{t-s}\left(\frac{s-t}{u-s}\varphi(s) + \frac{t-s}{u-s}\varphi(u)\right)\\ &= \frac{\varphi(u)-\varphi(s)}{u-s},\end{aligned}$$

obteniendo la primera desigualdad. Por otra parte, si invertimos los signos en la igualdad 1.2, sumamos $\varphi(u)$ y dividimos por $u-t$, obtenemos, siguiendo el mismo procedimiento,

$$\begin{aligned}\frac{\varphi(u)-\varphi(t)}{u-t} &\geq \frac{1}{u-t}\left(\varphi(u) - \frac{u-t}{u-s}\varphi(s) - \frac{t-s}{u-s}\varphi(u)\right)\\ &= \frac{1}{u-t}\left(\frac{u-t}{u-s}\varphi(u) - \frac{u-t}{u-s}\varphi(s)\right)\\ &= \frac{\varphi(u)-\varphi(s)}{u-s},\end{aligned}$$

obteniendo la desigualdad restante. □

Veamos finalmente la desigualdad de Jensen.

**Teorema 1.2.5** (Desigualdad de Jensen)**.** *Sea $\mu$ una medida de probabilidad sobre una $\sigma$-álgebra $\mathcal{A}$ en un conjunto $\Omega$. Si $f\colon \Omega \to \mathbb{R}$ es una función real integrable respecto a $\mu$, con $f(\Omega) \subset ]a,b[$, y $\varphi\colon ]a,b[ \to \mathbb{R}$ es convexa, entonces*

$$\varphi\left(\int_\Omega f\ d\mu\right) \leq \int_\Omega (\varphi \circ f)\ d\mu \tag{1.3}$$

*Además, si $\varphi$ es estrictamente convexa, se da la igualdad si y solo si $f$ es constante c.p.d.*

*Demostración.* Llamamos $t = \int_\Omega f\ d\mu$. Como $a < f(x) < b$ para todo $x \in \Omega$ y $\mu(\Omega) = 1$, se tiene

$$a = \int_\Omega a\ d\mu < \int_\Omega f\ d\mu < \int_\Omega b\ d\mu = b,$$

luego $a < t < b$. Tomamos ahora $s, u \in \mathbb{R}$ tales que $a < s < t < u < b$. Por el lema 1.2.4, se tiene que

$$\frac{\varphi(t)-\varphi(s)}{t-s} \leq \frac{\varphi(u)-\varphi(t)}{u-t},$$

en particular, existe $\beta = \sup_s \left\{\frac{\varphi(t)-\varphi(s)}{t-s}\colon a < s < t\right\}$. Además, $\beta$ es menor o igual que todos los cocientes de la derecha, para cualesquiera $t < u < b$, pues en caso contrario podríamos encontrar un $s < t$ para el cual no se verifica la desigualdad proporcionada por el lema. Por tanto,

$$\frac{\varphi(t)-\varphi(s)}{t-s} \leq \beta \leq \frac{\varphi(u)-\varphi(t)}{u-t} \quad \forall a < s < t < u < b. \tag{1.4}$$

Distinguimos casos en la desigualdad anterior:

- Si $a < s < t$, $\varphi(t) + \beta(s-t) \leq \varphi(s)$.

- Si $b < u < t$, $\beta(u-t) + \varphi(t) \leq \varphi(u)$.

Ambos casos, junto con la continuidad de $\varphi$ (es convexa en un abierto), nos permiten concluir que $\varphi(s) \geq \varphi(t) + \beta(s-t)$, para todo $a < s < b$.

Para cada $x \in \Omega$ podemos tomar $s = f(x) \in ]a, b[$, obteniendo, en la desigualdad anterior, que

$$\varphi(f(x)) - \varphi(t) - \beta(f(x) - t) \geq 0, \quad \forall x \in \Omega. \tag{1.5}$$

Como $\varphi$ es continua, $\varphi \circ f$ es medible, y por tanto podemos integrar respecto a $\mu$ ambos términos de la desigualdad anterior, obteniendo (usando que $\mu(\Omega) = 1$ y $t = \int_\Omega f \, d\mu$), que

$$\begin{aligned}
0 &\leq \int_\Omega (\varphi \circ f)(x) \, dx - \int_\Omega \varphi(t) \, dx - \int_\Omega \beta(f(x) - t) \, dx \\
&= \int_\Omega (\varphi \circ f)(x) \, dx - \varphi(t) - \beta\left(\int_\Omega f \, d\mu - t\right) \\
&= \int_\Omega (\varphi \circ f)(x) \, dx - \varphi\left(\int_\Omega f \, d\mu\right) - \beta\left(\int_\Omega f \, d\mu - \int_\Omega f \, d\mu\right) \\
&= \int_\Omega (\varphi \circ f)(x) \, dx - \varphi\left(\int_\Omega f \, d\mu\right),
\end{aligned}$$

obteniendo así la desigualdad buscada.

Supongamos ahora que $\varphi$ es estrictamente convexa y se da la igualdad, y supongamos que $f$ no es constante c.p.d. Si $t = \int_\Omega f \, d\mu$, entonces el conjunto $C = \{x \in \Omega \colon f(x) > t\}$ verifica $\mu(C) > 0$. Razonando de forma análoga a la del lema 1.2.4, es posible probar que la convexidad estricta implica que $\varphi(s) > \varphi(t) + \beta(s-t)$, para $a < s < b$ y $s \neq t$, luego la desigualdad 1.5 la podemos escribir como $\varphi(f(x)) > \varphi(t) + \beta(f(x) - t)$, para todo $x \in C$. Sin embargo, esto contradice que se haya dado la igualdad en la desigualdad de Jensen, pues dicha igualdad implica que $\varphi(f(x)) = \varphi(t) + \beta(f(x) - t)$ c.p.d, y $C$ es un conjunto de medida no nula donde no se da dicha igualdad. □

## 1.3. Problemas de optimización y algoritmos básicos

Los problemas de optimización aparecen en muchos campos del aprendizaje automático en general, y en particular, en la rama que vamos a tratar en este trabajo. Dentro de este tipo de problemas, la convexidad juega un papel fundamental, asegurando la globalidad de los óptimos encontrados. En esta sección estudiaremos estos problemas, proporcionando algunas herramientas genéricas para resolverlos.

### 1.3.1. Definiciones

**Definición 1.3.1.** Un *problema de minimización* es un problema de la forma

$$\min_{x \in \Omega} \quad f(x),$$

donde $f \colon \Omega \to \mathbb{R}$ se denomina *función objetivo*. De forma análoga se definen los *problemas de maximización*. Observemos que minimizar $f$ es equivalente a maximizar $-f$, luego podemos centrarnos únicamente en los problemas de minimización. Tanto los problemas de minimización como de maximización los denominaremos *problemas de optimización*.

Un *problema de minimización con restricciones* es un problema de minimización de la forma

$$\begin{aligned} \min_{x\in\Omega} \quad & f(x) \\ \text{sujeto a:} \quad & g_i(x) \leq 0, \quad i = 1, \dots, m \\ & h_i(x) = 0, \quad i = 1, \dots, p. \end{aligned}$$

La expresión *"sujeto a"* suele abreviarse por *"s.a."*, y las expresiones que figuran detrás se denominan *restricciones de desigualdad* y *restricciones de igualdad*, respectivamente. Las correspondientes funciones $g_i, h_j : \Omega \to \mathbb{R}$ se denominan *funciones restricción*, de desigualdad e igualdad, en cada caso. Análogamente se definen los *problemas de maximización con restricciones*, y ambos tipos de problemas conforman los *problemas de optimización con restricciones*.

Un punto $x \in \Omega$ se dice que es *viable* para un problema de optimización si satisface todas las restricciones del problema. Diremos que el problema es *viable* si admite puntos *viables*. En tal caso, se define el *valor óptimo* del problema como el ínfimo (supremo en problemas de maximización) en $\overline{\mathbb{R}} = [-\infty, +\infty]$ de las evaluaciones de la función objetivo en el conjunto de puntos viables. Diremos que el óptimo es *alcanzable* si existe un punto viable $x$ tal que $f(x)$ es el valor óptimo del problema. En tal caso, dicho $x$ es una *solución* del problema de optimización, y diremos que el problema *tiene solución*. Por último, diremos que $x \in \Omega$ es un *óptimo local* si es viable, y es un óptimo local de la función objetivo.

Un problema de minimización diremos que es *convexo* si tanto la función objetivo como las funcíones restricción de desigualdad son convexas, y las funciones restricción de igualdad son afintes. Análogamente, diremos que un problema de maximización es convexo si las funciones restricción de desigualdad son convexas, las funciones restricción de igualdad son afines y la función objetivo es cóncava. En tal caso, cada restricción define un conjunto convexo, y el conjunto de puntos viables, que es la intersección de dichos conjuntos, es también convexo, luego en este tipo de problemas de minimización se minimiza una función convexa sobre un conjunto convexo, por lo que las herramientas del análisis convexo pueden ser utilizadas.

Algunos problemas de optimización en el que la función objetivo y las restricciones satisfacen determinadas restricciones han sido ampliamente estudiados. Por ejemplo, cuando tanto la función objetivo como las funciones restricción son afines, el problema se conoce como *programación lineal*. Cuando la función objetivo es cuadrática y las restricciones son afines, el problema se conoce como *programación cuadrática*. Si el problema de optimización es matricial, sujeto a la restricción de ser semidefinida positiva y otras restricciones afines, el problema se conoce como *programación semidefinida*. Para este tipo de problemas se conocen diversos algoritmos capaces de encontrar una solución [3]. Nosotros nos centraremos en métodos generales válidos para la optimización sin restricciones o para la optimización con restricciones convexas arbitrarias.

### 1.3.2. Método del gradiente con proyecciones

Comenzamos analizando un método clásico para la minimización de funciones: el gradiente descendente. Es conocido que el gradiente de una función diferenciable tiene la dirección de la máxima pendiente en el grafo de la función, por lo que avanzando pequeñas cantidades en la dirección opuesta a la del gradiente conseguimos reducir el valor de la función. Este método iterativo es el que se conoce como gradiente descendente. La regla de actualización de este método iterativo, para encontrar un $x \in \mathbb{R}^d$ que minimice una función objetivo diferenciable $f : \mathbb{R}^d \to \mathbb{R}$ viene dada por $x_{t+1} = x_t - \eta \nabla f(x_t), t \in \mathbb{N} \cup \{0\}$, donde $\eta$ es la cantidad que se avanza en la dirección del gradiente, y se denomina *tasa de aprendizaje*. Dicho $\eta$

puede ser constante o puede ir adaptándose de acuerdo con las evaluaciones de la función objetivo. En el primer caso, la elección de un $\eta$ demasiado grande o demasiado pequeño puede conducir a malos resultados. El segundo caso requiere evaluar la función objetivo en cada iteración, lo que puede ser costoso computacionalmente.

Los fundamentos del gradiente descendente se basan en las siguientes ideas. Consideramos una función objetivo $f\colon \mathbb{R}^d \to \mathbb{R}$, $x \in \Omega$ y $v \in \mathbb{R}^d \setminus \{0\}$ una dirección arbitraria. Consideramos la función $g\colon \mathbb{R} \to \mathbb{R}$ dada por $g(\eta) = f(x + \eta \frac{v}{\|v\|})$. La tasa de variación o derivada direccional de $f$ en $x$ para la dirección $v$ viene dada por $g'(0) = \frac{1}{\|v\|}\langle \nabla f(x), v\rangle$. Aplicando la desigualdad de Cauchy-Schwarz, se tiene

$$-\|\nabla f(x)\| \leq \frac{1}{\|v\|}\langle \nabla f(x), v\rangle \leq \|\nabla f(x)\|,$$

y la igualdad en la desigualdad izquierda se alcanza cuando $v = -\nabla f(x)$, obteniendo así la tasa máxima de descenso. De la misma forma, la tasa de máximo ascenso se alcanza con $\nabla f(x)$.

Si el gradiente en $x$ es no nulo, y consideramos la aproximación de Taylor de primer orden para los puntos $x - \eta\nabla f(x)$ y $x$, se tiene

$$f(x - \eta\nabla f(x)) = f(x) - \eta\|\nabla f(x)\|^2 + o(\eta),$$

con $\lim_{\eta\to 0} |o(\eta)|/\eta = 0$, luego existe $\varepsilon > 0$ tal que si $0 < \delta < \varepsilon$, se tiene

$$\frac{o(\delta)}{\delta} < \|\nabla f(x)\|,$$

y por tanto,

$$f(x - \delta\nabla f(x)) - f(x) = \delta\left(-\|\nabla f(x)\|^2 + \frac{o(\delta)}{\delta}\right) < \delta(-\|\nabla f(x)\|^2 + \|\nabla f(x)\|^2) = 0,$$

luego $f(x-\delta\nabla f(x)) < f(x)$ para $0 < \delta < \varepsilon$, luego tenemos garantizado que para una tasa de aprendizaje adecuada el método puede descender en cada iteración. Observemos que la dirección del gradiente no es la única dirección de descenso válida, sino que lo anterior sigue siendo válido para cualquier dirección $v \in \mathbb{R}^d$ con $\langle \nabla f(x), v\rangle < 0$. La elección de otras direcciones de descenso, aunque no sean las de máxima pendiente, pueden proporcionar mejores resultados en problemas determinados.

Cuando trabajamos con problemas de optimización con restricciones, el método del gradiente no puede ser aplicado directamente, pues la regla de adaptación $x_{t+1} = x_t - \eta\nabla f(x_t)$ no garantiza que $x_{t+1}$ sea un punto viable. El método del gradiente con proyecciones solventa este problema, cuando el problema de optimización es convexo, añadiendo una proyección sobre el conjunto viable en la regla de adaptación, es decir, si $C$ es el conjunto convexo determinado por las restricciones, que supondremos también cerrado (esta condición se tiene cuando las funciones restricción son continuas, y la convexidad grantiza la continuidad en el interior del dominio), y $P_C$ es la proyección sobre dicho conjunto, entonces la regla de adaptación se convierte en $x_{t+1} = P_C(x_t - \eta\nabla f(x_t))$. Para confirmar la validez de este método, tenemos que ver que la dirección $v = P_C(x - \eta\nabla f(x)) - x$ es una dirección de descenso, lo cual, por lo razonado anteriormente, se consigue si $\langle \nabla f(x), v\rangle < 0$.

Llamamos $x_1 = x - \eta\nabla f(x)$. Entonces, $v = P_C(x_1) - x$- Notemos que $\langle \nabla f(x), v\rangle < 0 \iff \langle x_1 - x, P_C(x_1) - x\rangle = -\eta\langle \nabla f(x), v\rangle > 0$. Si el gradiente es no nulo y $x_1 \in C$, entonces, $\langle x_1 - x, x_1 - x\rangle = \|x_1 - x\|^2 > 0$. Si $x_1 \notin C$, entonces el teorema de la proyección convexa 1.1.4 asegura que el semiespacio $H = \{y \in \mathbb{R}^d\colon \langle x_1 - P_C(x_1), y - P_C(x_1)\rangle \leq 0$ contiene a $C$. En particular,

$$0 \geq \langle x_1 - P_C(x_1), x - P_C(x_1)\rangle = \langle x_1 - x, x - P_C(x_1)\rangle + \|x - P_C(x_1)\|^2.$$

En consecuencia, $\langle x_1 - x, P_C(x_1) - x\rangle \geq \|x - P_C(x_1)\|^2 \geq 0$. Además, la igualdad se da si y solo si $x = P_C(x_1)$, y en tal caso el algoritmo habrá convergido (observemos que esto ocurre cuando $x \in \operatorname{Fr} C$ y la dirección de descenso proporcionada por el gradiente apunta hacia fuera de $C$ y de forma ortogonal al hiperplano soporte). Por tanto, mientras las iteraciones del gradiente con proyecciones provoquen algún movimiento en los puntos obtenidos, escogiendo la tasa de aprendizaje adecuada, tenemos la garantía de poder descender en la función objetivo. En la figura 1.1 se comparan visualmente el gradiente con proyecciones y el gradiente descendente.

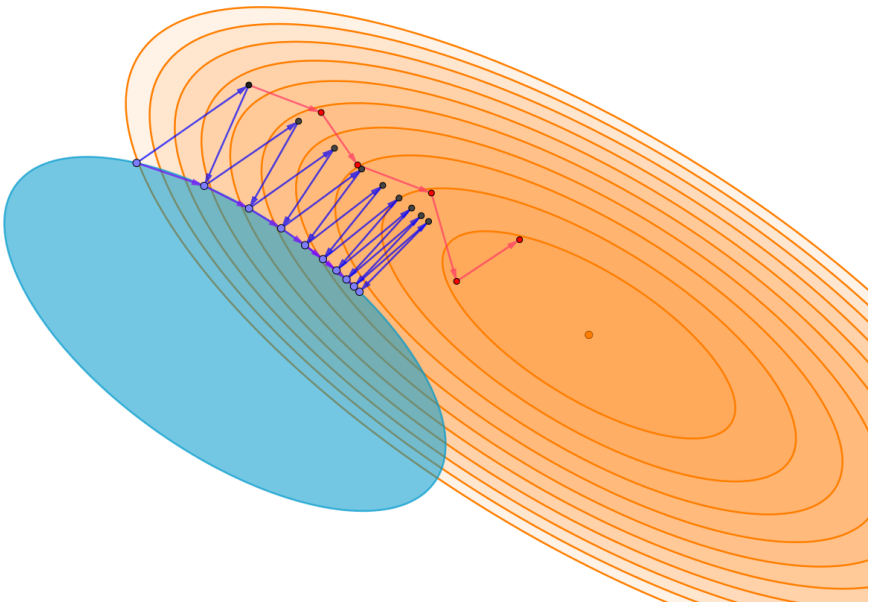

Figura 1.1: Las áreas naranjas sombreadas representan curvas de nivel de la función $f(x, y) = 2(x + y)^2 + 2y^2$ para valores naturales entre 0 y 10. El camino rojo muestra el comportamiento del gradiente descendente sin restricciones aplicado a la función $f$. El camino azul muestra el funcionamiento del gradiente con proyecciones sobre la elipse azul.

### 1.3.3. Método de las proyecciones iteradas

Si trabajamos con problemas convexos con restricciones, podemos encontrarnos con más de una restricción, de forma que no conozcamos una proyección explícita sobre el conjunto viable, que es la intersección de los conjuntos asociados a cada restricción. Un método que permite calcular un punto en dicha intersección, si conocemos las proyecciones sobre cada uno de los conjuntos que definen las restricciones, es el conocido como *método de las proyecciones iteradas*, que consiste en ir proyectando el punto sucesivas veces en cada uno de los conjuntos asociados a cada restricción. Dicha sucesión de proyecciones converge a la intersección, por lo que este método iterativo puede utilizarse para forzar la satisfacción de las restricciones en un problema convexo. Veamos que efectivamente el método de las proyecciones iteradas converge. Lo vamos a ver para dos restricciones. El caso general se prueba siguiendo el mismo razonamiento.

**Teorema 1.3.1** (Convergencia de las proyecciones iteradas)**.** *Sean $C, D \subset \mathbb{R}^d$ conjuntos convexos cerrados y sean $P_C, P_D \colon R^d \to R^d$ las proyecciones sobre $C$ y $D$, respectivamente. Supongamos que $x_0 \in C$ y construimos las sucesiones $\{x_n\}$ e $\{y_n\}$ dadas por $y_n = P_D(x_n)$ y $x_{n+1} = P_C(y_n)$, para cada $n \in \mathbb{N}$.*

*Entonces, si $C \cap D \neq \emptyset$, ambas sucesiones convergen a un punto $x^* \in C \cap D$.*

*Demostración.* Fijamos $\overline{x} \in C \cap D$ arbitrario. Si existe algún $k \in \mathbb{N}$ tal que $x_k \in C \cap D$ o $y_k \in C \cap D$, entonces $x_n = y_n = x_k \in C \cap D$, para todo $n > k$, lo que finalizaría la prueba. Por tanto, a partir de ahora supondremos que $x_n, y_n \notin C \cap D$ para todo $n \in \mathbb{N}$.

Observemos que, como $y_n = P_D(x_n)$ para todo $n \in \mathbb{N}$, el teorema de la proyección convexa 1.1.4 nos dice que el semiespacio $\{z \in \mathbb{R}^d \colon \langle x_n - y_n, z - y_n \rangle \leq 0\}$ contiene a $D$. Aplicando esto a $\overline{x} \in C \cap D \subset D$, se tiene que

$$\begin{aligned} \|x_n - \overline{x}\|^2 &= \|x_n - y_n + y_n - \overline{x}\|^2 \\ &= \|x_n - y_n\|^2 + \|y_n - \overline{x}\|^2 + 2\langle x_n - y_n, y_n - \overline{x} \rangle \\ &= \|x_n - y_n\|^2 + \|y_n - \overline{x}\|^2 - 2\langle x_n - y_n, \overline{x} - y_n \rangle \\ &\geq \|x_n - y_n\|^2 + \|y_n - \overline{x}\|^2. \end{aligned}$$

Por tanto,

$$\|y_n - \overline{x}\|^2 \leq \|x_n - \overline{x}\|^2 - \|y_n - x_n\|^2 \leq \|x_n - \overline{x}\|^2 \quad \forall n \in \mathbb{N}. \tag{1.6}$$

Análogamente, como $x_{n+1} = P_C(y_n)$, se tiene que el semiespacio $\{z \in \mathbb{R}^d \colon \langle y_n - x_{n+1}, z - x_{n+1} \rangle \leq 0\}$ contiene a $C$, y razonando como en la expresión anterior se deduce que

$$\|x_{n+1} - \overline{x}\|^2 \leq \|y_n - \overline{x}\|^2 - \|x_{n+1} - y_n\|^2 \leq \|y_n - \overline{x}\|^2 \quad \forall n \in \mathbb{N}. \tag{1.7}$$

En particular, se tiene que $\|x_n - \overline{x}\| \leq \|x_0 - \overline{x}\|$ y $\|y_n - \overline{x}\| \leq \|x_0 - \overline{x}\|$, para cada $n \in \mathbb{N}$, y en consecuencia $\{x_n\}$ e $\{y_n\}$ están acotadas. Por tanto, $\{x_n\}$ admite una parcial convergente a un punto $x^* \in C$, por ser $C$ cerrado y $\{x_n\} \subset C$. Veamos que también $x^* \in D$, y que es el límite de las sucesiones $\{x_n\}$ e $\{y_n\}$.

De las expresiones 1.6 y 1.7 se deduce que la sucesión $\{\|z_n - \overline{x}\|\}$, donde $z_{2k} = x_k$ y $z_{2k+1} = y_k$, para $k \in \mathbb{N} \cup \{0\}$, es decreciente. Como además está minorada, converge. Las sucesiones $\{\|x_n - \overline{x}\|\}$ e $\{\|y_n - \overline{x}\|\}$ son parciales de la sucesión anterior, luego convergen al mismo límite, que llamaremos $L$. Si tomamos límites superiores en 1.6, obtenemos que $L^2 \leq L^2 - \limsup\{\|y_n - x_n\|^2\} \leq L^2$, luego $\limsup\{\|y_n - x_n\|^2\} = 0$. Análogamente, tomando límites inferiores, se deduce que $\liminf\{\|y_n - x_n\|^2\} = 0$, luego $\|y_n - x_n\| \to 0$. Razonando igualmente con la expresión 1.7, se obtiene que también $\|x_{n+1} - y_n\| \to 0$. Como $d(x_n, D) = d(x_n, P_D(x_n)) = d(x_n, y_n) = \|x_n - y_n\| \to 0$ y $x^*$ es el límite de una sucesión parcial de $\{x_n\}$, se deduce que $d(x^*, D) = 0$, y como $D$ es cerrado, se tiene $x^* \in D$, luego $x^* \in C \cap D$.

Como $\overline{x}$ era arbitrario, podemos tomar $\overline{x} = x^* \in C \cap D$. Entonces, por lo ya visto para $\overline{x}$, se tiene que las sucesiones $\{\|x_n - x^*\|\}$ e $\{\|y_n - x^*\|\}$ son decrecientes, luego convergen. Como $x^*$ era el límite de una parcial de $\{x_n\}$, se deduce que $\{\|x_n - x^*\|\} \to 0$. Finalmente, $\|y_n - x^*\| \leq \|y_n - x_n\| + \|x_n - x^*\| \to 0$, concluyendo así con la convergencia de las proyecciones iteradas.

□

Para concluir, es interesante destacar que el teorema anterior admite una versión cuando la intersección es vacía. En tal caso, es posible probar que, si hay puntos donde se alcanza la distancia entre los conjuntos $C$ y $D$, las sucesiones convergerán, cada una en su conjunto, a uno de esos puntos [6]. También, notemos que, cuando $C \cap D \neq \emptyset$, el límite de las sucesiones no es necesariamente la proyección sobre la intersección. Sin embargo, un razonamiento mediante hiperplanos soporte similar al utilizado en el método del gradiente con proyecciones permite ver que la dirección que apunta al límite es también una dirección de descenso. En la figura 1.2 se muestra gráficamente el funcionamiento del método de las proyecciones iteradas en ambos casos.

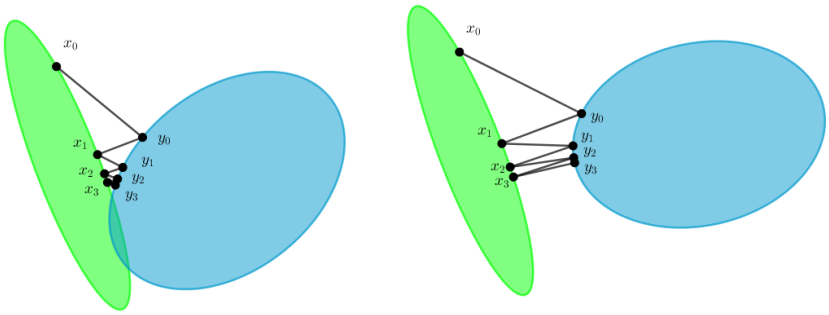


Figura 1.2: Método de las proyecciones iteradas. La segunda imagen muestra el desarrollo del método cuando los conjuntos no se cortan.

### 1.3.4. Subgradientes

Recordemos que cuando $f\colon \Omega \to \mathbb{R}$, con $\Omega$ convexo y abierto, es convexa y diferenciable, se verifica que $f(x) \geq \langle \nabla f(x_0), x - x_0 \rangle$ para cualesquiera $x, x_0 \in \Omega$. En general, si $f\colon A \to \mathbb{R}$ es una función arbitraria, cualquier vector $v \in \mathbb{R}^d$ que verifique, para $x_0 \in A$, que $f(x) \geq f(x_0) + \langle v, x - x_0 \rangle$ para todo $x \in A$ se denomina un *subgradiente* de $f$ en $x_0$. Una función puede no tener, o tener más de un subgradiente. El conjunto de los subgradientes se $f$ en el punto $x_0$ se denotará por $\partial f(x_0)$ (o $\partial f(x_0)/\partial x$, si es necesario especificar la variable).

El subgradiente tiene un comportamiento similar al del gradiente, y por ello se utilizan como método de descenso cuando la función no es diferenciable. Es importante notar que no todos los subgradientes llevan siempre una dirección de descenso, ni necesariamente han de existir. En el caso de las funciones convexas sí que podemos garantizar la existencia de subgradientes en todos los puntos.

**Proposición 1.3.2.** *Sea $\Omega \subset \mathbb{R}^d$ un abierto convexo. Una función $f\colon \Omega \to \mathbb{R}$ es convexa si y solo si admite subgradientes en todo punto de $\Omega$.*

*Demostración.* Por un lado, $f$ es convexa si y solo si $\mathrm{Epi}(f)$ es convexo. Por ser $\Omega$ abierto, $\mathrm{Epi}(f)$ tiene interior no vacío en $\mathbb{R}^{d+1}$. Por otro lado, $v$ es un subgradiente de $f$ en $x_0$ si y solo si $(v, -1)$ define un hiperplano soporte en $\mathrm{Epi}(f)$ sobre el punto frontera $(x_0, f(x_0))$. Esto se debe a que $(x, t) \in \mathrm{Epi}(f)$, si y solo si $t \geq f(x) \geq f(x_0) + \langle v, x - x_0 \rangle$, lo cual ocurre si y solo si $\langle v, x \rangle - t \leq f(x_0) + \langle v, x_0 \rangle \iff \langle (v, -1), (x, t) \rangle f(x_0) + \langle v, x_0 \rangle$. Gracias a estas caracterizaciones, el teorema del hiperplano soporte 1.1.3 garantiza el resultado buscado. □

Por tanto, sobre funciones convexas podemos tomar subgradientes en todo punto e iterar de forma análoga al gradiente descendente o al gradiente con proyecciones. Como ya se ha dicho, el subgradiente puede no llevar una dirección de descenso. Por ello, al aplicar este método se suele almacenar el mejor valor obtenido. Aunque exista más de un subgradiente, solo necesitamos calcular uno para aplicar el método del subgradiente. En los algoritmos que trataremos el cálculo de subgradientes será sencillo. Por ejemplo, si $f$ es diferenciable en $x$, podemos tomar el subgradiente $\nabla f(x)$ (de hecho, este es el único subgradiente en este caso). El otro caso de interés que trataremos será el cálculo del subgradiente de máximos de funciones convexas diferenciables.

**Proposición 1.3.3.** *Sea $\Omega \subset \mathbb{R}^d$ abierto y convexo, y $f_i\colon \Omega \to \mathbb{R}$, con $i \in \{1, \ldots, r\}$, funciones convexas*

*diferenciables. Sea* $f\colon \Omega \to \mathbb{R}$ *definida fomo* $f(x) = \max_i f_i(x)$. *Dado* $x\Omega$, *tomamos* $j \in \{1, \ldots, r\}$ *tal que* $f(x) = f_j(x)$. *Entonces,* $\nabla f_j(x) \in \partial f(x)$.
*Demostración.* Como $f_j$ es convexa se tiene que, para todo $y \in \Omega$, $f_j(y) \geq f_j(x) + \langle y - x, \nabla f_j(x)\rangle$. Como $f(x) = f_j(x)$ y $f(y) \geq f_j(y)$, se concluye que

$$f(y) \geq f(x) + \langle y - x, \nabla f_j(x)\rangle.$$

□

# Capítulo 2

# Análisis matricial

En el aprendizaje de métricas de distancia, las matrices tendrán un papel fundamental, pues serán la estructura que definirá las distancias y sobre la que se aplicarán los métodos de optimización estudiados en el capítulo anterior. Dentro del conjunto de todas las matrices, las matrices semidefinidas serán de aún mayor importancia, por lo que para comprender mejor los problemas de aprendizaje con los que trataremos será necesario profundizar en algunas de sus numerosas propiedades.

Este capítulo profundiza en el estudio de las matrices, partiendo de los resultados más conocidos de diagonalización en el álgebra lineal. Desde esta base, se introducirá un producto escalar en el espacio de las matrices, convirtiéndolas así en un espacio de Hilbert, añadiendo de esta forma propiedades métricas y topológicas que analizaremos. A continuación nos centraremos en las matrices definidas positivas, estudiando varios teoremas de descomposición que serán de gran utilidad. También veremos cómo el producto escalar que hemos añadido permite demostrar un teorema de proyección que motivará muchos de los algoritmos que se estudiarán más adelante. Por último, analizaremos cómo trabajar con problemas de optimización basados en matrices, haciendo especial hincapié en determinados problemas de optimización que se pueden resolver mediante vectores propios.

## 2.1. Preliminares

Nos centraremos en el estudio de las matrices con entradas reales, pues el problema que vamos a tratar será en variable real, si bien muchos de los resultados que vamos a ver son extensibles al caso complejo. Introducimos en primer lugar la notación que utilizaremos para las matrices a lo largo de este trabajo.

Notaremos el espacio de las matrices reales de dimensión $d' \times d$ como $\mathcal{M}_{d' \times d}(\mathbb{R})$. Cuando $d' = d$, entonces abreviaremos, notando el espacio de las matrices cuadradas de orden $d$ como $\mathcal{M}_d(\mathbb{R})$. Una matriz $A \in \mathcal{M}_{d' \times d}(\mathbb{R})$ también la podremos expresar, aludiendo a sus entradas, como $A = (A_{ij})_{\substack{i=1,\dots,d' \\ j=1,\dots,d}}$, donde $A_{ij} \in \mathbb{R}$ representa la entrada en la fila $i$-ésima y columna $j$-ésima. También usaremos la notación $A_{\cdot j}$ para aludir a la columna $j$-ésima completa de la matriz, vista como vector, y análogamente $A_{i\cdot}$ para la fila $i$-ésima. Los vectores $v = (v_1, \dots, v_d) \in \mathbb{R}^d$ los trataremos como matrices columna.

Dada $A = (A_{ij}) \in \mathcal{M}_{d' \times d}(\mathbb{R})$, notaremos la matriz traspuesta de $A$ como $A^T = (A_{ji}) \in \lceil \times \lceil'(\mathbb{R})$. El conjunto de las matrices simétricas de orden $d$ lo notaremos por $S_d(\mathbb{R}) = \{A \in \mathcal{M}_d(\mathbb{R})\colon A = A^T\}$. Al conjunto de las matrices antisimétricas lo expresaremos como $A_d(\mathbb{R}) = \{A \in \mathcal{M}_d(\mathbb{R})\colon A = -A^T$. El conjunto de las matrices regulares de orden $d$ o grupo lineal de orden $d$ lo escribiremos como

$$\begin{aligned} \mathrm{GL}_d(\mathbb{R}) &= \{A \in \mathcal{M}_d(\mathbb{R})\colon \exists A^{-1} \in \mathcal{M}_d(\mathbb{R})\colon AA^{-1} = A^{-1}A = I\} \\ &= \{A \in \mathcal{M}_d(\mathbb{R})\colon \det(A) \neq 0\} = \{A \in \mathcal{M}_d(\mathbb{R})\colon r(A) = d\}, \end{aligned}$$

donde las operaciones $r$ y det hacen referencia al rango y al determinante, respectivamente. También utilizaremos el operador traza, que notaremos por tr. Una matriz $A \in \mathcal{M}_d(\mathbb{R})$ diremos que es ortogonal si es regular y $A^T = A^{-1}$. El conjunto de las matrices ortogonales lo notaremos por $O_d(\mathbb{R})$.

Una matriz $M \in S_d(\mathbb{R})$ diremos que es semidefinida positiva si se verifica que $x^T M x \geq 0$ para todo $x \in \mathbb{R}^d$. Si además se tiene que $x^T M x = 0 \iff x = 0$, diremos que $M$ es definida positiva. Notaremos a estos conjuntos por $\mathcal{M}_d(\mathbb{R})_0^+$ y $\mathcal{M}_d(\mathbb{R})^+$, respectivamente. Las matrices semidefinidas positivas (resp. definidas positivas) se caracterizan por tener todos sus valores propios no negativos (resp. positivos). Análogamente se definen las matrices semidefinidas y definidas negativas, y se notan por $\mathcal{M}_d(\mathbb{R})_0^-$ y $\mathcal{M}_d(\mathbb{R})^-$.

Es conocido que, fijada una base en $\mathbb{R}^d$, las matrices de dimensión $d' \times d$ se identifican con el conjunto de aplicaciones lineales de $\mathbb{R}^d$ en $\mathbb{R}^{d'}$ de forma biunívoca. Por tanto, si es necesario, una matriz $L \in \mathcal{M}_{d' \times d}(\mathbb{R})$ la veremos como una aplicación lineal $L\colon \mathbb{R}^d \to \mathbb{R}^{d'}$, notando ambas de la misma forma. También es conocido que las matrices simétricas se identifican con operadores autoadjuntos si la base fijada es ortonormal. Pero las matrices simétricas $A \in S_d(\mathbb{R})$ además pueden identificarse con formas bilineales simétricas, esto es, aplicaciones de la forma $g_A\colon \mathbb{R}^d \to \mathbb{R}^d$ dadas por $g_A(x, y) = x^T A y$, que además son semidefinidas o definidas positivas si la matriz $A$ lo es. De nuevo, según las circunstancias, podremos ver las matrices simétricas como aplicaciones, si es necesario.

## 2.2. Las matrices como espacio de Hilbert

Sobre el conjunto de las matrices de determinada dimensión tenemos definida una operación de suma, y además tenemos definido un producto entre matrices de órdenes $d \times r$ y $r \times n$, que cuando nos restringimos a matrices cuadradas, se convierte en un producto interno, que junto con la suma da al espacio vectorial de las matrices una estructura de anillo no conmutativo. Con estas operaciones solo podemos obtener propiedades algebraicas sobre las matrices. Si queremos desarrollar una teoría geométrica sobre ellas tendremos que introducir una topología adecuada. La topología que introduciremos permitirá tratar el espacio de matrices $\mathcal{M}_{d' \times d}(\mathbb{R})$ como un espacio de Hilbert idéntico a $\mathbb{R}^{d' \times d}$. Esta identificación será únicamente a nivel de espacios de Hilbert, de forma que con las herramientas matriciales como el producto matricial o los valores propios podremos estudiar propiedades que en general no se presentan en $\mathbb{R}^{d' \times d}$ como espacio vectorial.

**Definición 2.2.1.** Se define el *producto escalar de Frobenius* en el espacio de matrices de orden $d' \times d$ como la aplicación $\langle \cdot, \cdot \rangle_F\colon \mathcal{M}_{d' \times d}(\mathbb{R}) \times \mathcal{M}_{d' \times d}(\mathbb{R}) \to \mathbb{R}$ dada por

$$\langle A, B \rangle_F = \sum_{i=1}^{d} \sum_{j=1}^{d'} A_{ij} B_{ij} = \operatorname{tr}(A^T B).$$

Se define la *norma de Frobenius* en el espacio de matrices de orden $d' \times d$ como la aplicación $\| \cdot \|_F\colon \mathcal{M}_{d' \times d}(\mathbb{R}) \to \mathbb{R}_0^+$ dada por

$$\|A\|_F = \sqrt{\langle A^T, A \rangle} = \sqrt{\sum_{i=1}^{d} \sum_{j=1}^{d'} A_{ij}^2} = \sqrt{\operatorname{tr}(A^T A)}$$

Se tiene que $(\mathcal{M}_{d' \times d}(\mathbb{R}), \langle \cdot, \cdot \rangle_F)$ es un espacio de Hilbert, y su producto escalar es el mismo que si calculáramos el producto escalar usual de la matriz vista como vector en $\mathbb{R}^{d' \times d}$, añadiendo las filas al vector una detrás de otra.

La norma de Frobenius es, por tanto, idéntica a la norma euclidea en $\mathbb{R}^{d'\times d}$ realizando la misma identificación entre matrices y vectores. Vista como norma matricial, verifica propiedades adicionales. Por ejemplo, si consideramos matrices cuadradas, tenemos que la norma de Frobenius es submultiplicativa, esto es, $\|AB\|_F \leq \|A\|_F\|B\|_F$, para $A, B \in \mathcal{M}_d(\mathbb{R})$. Esto se deduce aplicando las definiciones de norma de Frobenius y producto matricial, y aplicando la desigualdad de Cauchy-Schwarz para el producto escalar en $\mathbb{R}^d$. De hecho, la desigualdad se verifica para matrices no cuadradas, siempre que las dimensiones permitan multiplicarlas, y tomando en cada caso la norma de Frobenius en el espacio adecuado.

Muchas de las normas que se definen para las matrices cuadradas tienen la propiedad de que son *inducidas* por una norma vectorial, esto es, la norma $|||\cdot|||$ en $\mathcal{M}_d(\mathbb{R})$ está inducida por la norma $\|\cdot\|$ en $\mathbb{R}^d$ si para toda $A \in \mathcal{M}_d(\mathbb{R})$ se tiene

$$|||A||| = \sup\{\|Ax\| \colon x \in \mathbb{R}^d, \|x\| = 1\} = \sup\left\{\frac{\|Ax\|}{\|x\|} \colon x \in \mathbb{R}^d \setminus \{0\}\right\}.$$

En este caso, se tiene que la norma de Frobenius no está inducida por ninguna norma vectorial, para $d \geq 2$. Observemos que todas las normas inducidas por una norma vectorial han de verificar $|||I||| = 1$, lo cual no ocurre con la norma de Frobenius, pues $\|I\|_F = \sqrt{d} > 1$.

Además de las ya comentadas, es importante destacar las siguientes propiedades de la norma de Frobenius.

**Proposición 2.2.1.**

a) *Para cada* $A \in \mathcal{M}_{d'\times d}(\mathbb{R})$, $\|A\|_F = \|A^T\|_F$.

b) *Para cada* $A \in \mathcal{M}_{d'\times d}(\mathbb{R})$, $\|A\|_F = \sqrt{\operatorname{tr}(AA^T)}$

c) *Si* $U \in O_d(\mathbb{R}), V \in O_{d'}(\mathbb{R})$ *y* $A \in \mathcal{M}_{d'\times d}(\mathbb{R})$, *entonces* $\|AU\|_F = \|VA\|_F = \|VAU\|_F = \|A\|_F$.

d) *Si* $A \in S_d(\mathbb{R})$, *entonces* $\|A\|_F^2 = \sum_{i=1}^d \lambda_i^2$, *donde* $\lambda_1, \ldots, \lambda_d$ *son los valores propios de* $A$.

e) *Si* $A \in S_d(\mathbb{R})$, *entonces,* $\rho(A) \leq \|A\|_F \leq \sqrt{d}\rho(A)$, *donde* $\rho(A) = \max\{|\lambda| \colon \lambda \in \mathbb{R}$ *es valor propio de* $A\}$ *es el* radio espectral *de* $A$.

*Demostración.*

a) Es evidente.

b) Consecuencia de la invarianza de la traza por permutaciones cíclicas.

c) Se tiene

$$\|AU\|_F^2 = \operatorname{tr}((AU)^T(AU)) = \operatorname{tr}(U^TA^TAU) = \operatorname{tr}(UU^TA^TA) = \operatorname{tr}(A^TA) = \|A\|_F^2$$
$$\|VA\|_F^2 = \operatorname{tr}((VA)^T(VA)) = \operatorname{tr}(A^TV^TVA) = \operatorname{tr}(A^TA) = \|A\|_F^2$$

d) Si $A \in S_d(\mathbb{R})$, existe una matrix $U \in O_d(\mathbb{R})$ tal que $A = UDU^T$ y $D = \operatorname{diag}(\lambda_1, \ldots, \lambda_d)$ es la matriz diagonal con los valores propios de $A$. Entonces, por la propiedad anterior,

$$\|A\|_F^2 = \|UDU^T\|_F^2 = \|D\|_F^2 = \sum_{i,j=1}^d D_{ij}^2 = \sum_{i=1}^d \lambda_i^2.$$

e) Si $D = \operatorname{diag}(\lambda_1, \ldots, \lambda_d)$ es la matriz diagonal con los valores propios de $A$, se tiene que

$$\rho(A) = \|(\lambda_1, \ldots, \lambda_d)\|_\infty \leq \|(\lambda_1, \ldots, \lambda_d)\|_2 \leq \sqrt{d}\|(\lambda_1, \ldots, \lambda_d)\|_\infty = \sqrt{d}\rho(A).$$

El resultado se tiene al observar que $\|(\lambda_1, \dots, \lambda_d)\|_2 = \|D\|_F = \|A\|_F$.

□

Para concluir, observemos que, teniendo las matrices identificadas como elementos de un espacio vectorial de dimensión finita con producto escalar, todas las teorías métricas pueden ser desarrolladas de la misma forma que las teorías en un espacio $\mathbb{R}^d$. En particular, todos los resultados vistos en el capítulo anterior pueden aplicarse sobre las matrices.

## 2.3. Matrices semidefinidas positivas: teoremas de descomposición y proyección

### 2.3.1. El cono de las matrices semidefinidas positivas

En esta sección nos centraremos en el estudio de las matrices semidefinidas positivas. Comenzaremos viendo su estructura algebraica como conjunto. Sean $A, B \in \mathcal{M}_d(\mathbb{R})_0^+$ y $\alpha_1, \alpha_2 \in \mathbb{R}_0^+$. Entonces, dado $x \in \mathbb{R}^d$,

$$x^T(\alpha_1 A + \alpha_2 B)x = \alpha_1(x^T A x) + \alpha_2(x^T B x) \geq 0,$$

luego $\alpha_1 A + \alpha_2 B \in \mathcal{M}_d(\mathbb{R})_0^+$. Por tanto, el conjunto de las matrices semidefinidas positivas tiene estructura de cono, y en particular es convexo. Viendo este conjunto como subconjunto de las matrices simétricas, y con la topología inducida por estas, tenemos que el cono de matrices semidefinidas positivas verifica también las siguientes propiedades:

- Es cerrado. Podemos ver

$$\mathcal{M}_d(\mathbb{R})_0^+ = \{M \in S_d(\mathbb{R})\colon x^T M x \geq 0 \quad \forall x \in \mathbb{R}^d\} = \bigcap_{x \in \mathbb{R}^d} \{M \in S_d(\mathbb{R})\colon x^T M x \geq 0\}.$$

  Los elementos en la intersección son semiespacios cerrados, dentro del conjunto de las matrices simétricas, pues la aplicación $M \mapsto x^T M x$ es lineal en $M$, fijado $x \in \mathbb{R}^d$. Por tanto, la intersección es cerrada.

- Es puntiagudo. Observemos que $-\mathcal{M}_d(\mathbb{R})_0^+ = \mathcal{M}_d(\mathbb{R})_0^-$. Si $M \in \mathcal{M}_d(\mathbb{R})_0^+ \cap \mathcal{M}_d(\mathbb{R})_0^-$, entonces se tiene que todos sus valores propios son no negativos y no positivos, luego todos sus valores propios son 0, y esto solo es posible si $M = 0$.

- Es sólido. Concretamente, su interior es $\mathcal{M}_d(\mathbb{R})^+$, que es no vacío. Para probarlo utilizaremos el siguiente resultado.

**Proposición 2.3.1** (Propiedades de regularización)**.**

a) *Sea $M \in S_d(\mathbb{R})$. Entonces, existe $\varepsilon > 0$ tal que $M + \varepsilon I \in \mathcal{M}_d(\mathbb{R})^+$.*

b) *Sea $M \in \mathcal{M}_d(\mathbb{R})_0^+$. Entonces, para todo $\varepsilon > 0$, se tiene que $M + \varepsilon I \in \mathcal{M}_d(\mathbb{R})^+$.*

*Demostración.* Sea $M \in S_d(\mathbb{R})$. Entonces, todos sus valores propios son reales, y son raíces del polinomio característico $p(\lambda) = \det(M - \lambda I)$. Si llamamos $\lambda_1 \leq \cdots \leq \lambda_d$ a los valores propios de $M$, para cada $\varepsilon \in \mathbb{R}$ se tiene que las raíces del polinomio $q(\lambda) = p(\lambda - \varepsilon)$ son $\lambda_1 + \varepsilon \leq \cdots \leq \lambda_d + \varepsilon$. Además, el polinomio $q$ es justo el polinomio característico asociado a $M + \varepsilon I$, pues $\det((M + \varepsilon - I) - \lambda I) = \det(M - (\lambda - \varepsilon)I) = p(\lambda\varepsilon) = q(\lambda)$.

Por tanto, si tomamos $\varepsilon > \max\{-\lambda_1, 0\} \geq 0$, todos los valores propios de $M + \varepsilon I$ serán positivos, y por tanto $M + \varepsilon I \in \mathcal{M}_d(\mathbb{R})^+$. Si se tenía $M \in \mathcal{M}_d(\mathbb{R})_0^+$, entonces conseguimos que los valores propios sean positivos para cualquier $\varepsilon > 0$. □

La propiedad anterior es muy interesante desde el punto de vista computacional, pues en ocasiones los errores de precisión en los cálculos con matrices hacen que se pierda la condición de ser definida positiva. El resultado anterior nos dice que podemos recuperarla añadiendo un valor positivo a la diagonal de la matriz, siendo tan pequeño como queramos en el caso de tener una matriz semidefinida. De aquí que se enuncien como propiedades de regularización. Con estas propiedades vamos a terminar de ver que el interior del conjunto de matrices semidefinidas positivas son las matrices definidas positivas (viéndolas dentro de las matrices simétricas).

**Corolario 2.3.2.** *$\mathcal{M}_d(\mathbb{R})_0^+$ es un cono propio con interior $\mathcal{M}_d(\mathbb{R})^+$.*

*Demostración.* Ya hemos visto que $\mathcal{M}_d(\mathbb{R})_0^+$ es un cono cerrado y puntiagudo. Queda comprobar que su interior es el conjunto de las matrices definidas positivas. Notamos a las bolas inducidas por la norma de Frobenius sobre las matrices simétricas como $B(M, r) = \{A \in S_d(\mathbb{R}) \colon \|M - A\|_F < r\}$, para cada $M \in S_d(\mathbb{R})$ y $r > 0$.

Veamos que $(\mathcal{M}_d(\mathbb{R})_0^+)^{\circ} \subset \mathcal{M}_d(\mathbb{R})^+$. Sea $M \in (\mathcal{M}_d(\mathbb{R})_0^+)^{\circ}$. Entonces, existe $\varepsilon > 0$ tal que $B(M, 2\varepsilon\sqrt{d}) \subset \mathcal{M}_d(\mathbb{R})_0^+$. Sea $A = M - \varepsilon I$. Se tiene que $\|M - A\|_F = \|\varepsilon I\|_F = \varepsilon\sqrt{d}$, luego $A \in B(M, 2\varepsilon\sqrt{d}) \subset \mathcal{M}_d(\mathbb{R})_0^+$. Por la proposición anterior, $M = A + \varepsilon I \in \mathcal{M}_d(\mathbb{R})^+$.

Veamos que $\mathcal{M}_d(\mathbb{R})^+$ es abierto. El criterio de Sylvester nos dice que $M \in \mathcal{M}_d(\mathbb{R})^+$ si y solo si todos sus menores principales son positivos, esto es, si y solo si $\det(M_k) > 0$ para cada $k \in \{1, \ldots, d\}$, donde $M_k \in \mathcal{M}_k(\mathbb{R})$ es la matriz de orden $k$ cuyas entradas coinciden con las de $M$ hasta dicha dimensión. En consecuencia,

$$\mathcal{M}_d(\mathbb{R})^+ = \{M \in S_d(\mathbb{R}) \colon \det(M_k) > 0 \quad \forall k = 1, \ldots, d\} = \bigcap_{k=1}^{d} \{M \in \mathcal{S}_d(\mathbb{R}) \colon \det(M_k) > 0\}.$$

Por la continuidad del determinante, tenemos una intersección finita de conjuntos abiertos en $S_d(\mathbb{R})$ (son abiertos en $\mathcal{M}_d(\mathbb{R})$, intersecados con $S_d(\mathbb{R})$). Por tanto, $\mathcal{M}_d(\mathbb{R})^+$ es abierto.

Como $\mathcal{M}_d(\mathbb{R})^+ \subset \mathcal{M}_d(\mathbb{R})_0^+$ y $\mathcal{M}_d(\mathbb{R})^+$ es abierto, tomando interiores se deduce la inclusión restante. □

Puesto que hemos probado que las matrices semidefinidas positivas son un cono propio sobre las matrices simétricas, tenemos definidas sobre estas la relación de orden $\preceq$, dada por $A \preceq B \iff B - A \in \mathcal{M}_d(\mathbb{R})_0^+$. Análogamente, tenemos el orden estricto dado por $A \prec B \iff B - A \in \mathcal{M}_d(\mathbb{R})^+$. Estos órdenes se denominan órdenes de *Löwner* Veamos algunas de sus principales propiedades.

**Proposición 2.3.3** (Propiedades del orden de las matrices semidefinidas)**.** *Supongamos $A, B, M \in S_d(\mathbb{R})$.*

a) *$M \in \mathcal{M}_d(\mathbb{R})_0^+ \iff M \succeq 0$ y $M \in \mathcal{M}_d(\mathbb{R})^+ \iff M \succ 0$.*

b) *$A \preceq B \iff x^T A x \leq x^T B x$ para todo $x \in \mathbb{R}^d$.*

c) *$A \prec B \iff x^T A x < x^T B x$ para todo $x \in \mathbb{R}^d \setminus \{0\}$.*

d) *Si $A \preceq B$, entonces $C^T A C \preceq C^T B C$, para todo $C \in \mathcal{M}_d(\mathbb{R})$.*

e) *Si $A \prec B$, entonces $C^T A C \prec C^T B C$ para todo $C \in \mathrm{GL}_d(\mathbb{R})$.*

*Demostración.*

a) Es evidente.

b) $A \preceq B \iff B - A \succeq 0 \iff x^T(B-A)x \geq 0$ para todo $x \in \mathbb{R}^d \iff x^TBx \geq x^TAx$ para todo $x \in \mathbb{R}^d$.

c) La prueba es análoga a la anterior.

d) Basta ver que $B - A \succeq 0 \implies C^T(B-A)C \succeq 0$:

$$B-A \succeq 0 \implies (Cx)^T(B-A)(Cx) \geq 0 \quad \forall x \in \mathbb{R}^d \implies x^T(C^T(B-A)C)x \geq 0 \quad \forall x \in \mathbb{R}^d \implies C^T(B-A)C \succeq 0.$$

e) La prueba es análoga a la anterior, teniendo en cuenta que $Cx \neq 0$ para todo $x \neq 0$, pues $C$ es regular.

□

### 2.3.2. Teoremas de descomposición

Los próximos resultados que veremos serán teoremas de descomposición para matrices semidefinidas positivas, o basadas en estas. Estos resultados nos permitirán, por un lado, proporcionar varias alternativas para parametrizar el problema que trataremos en el capítulo 5. Por otra parte, estos teoremas de descomposición proporcionan la base para muchos algoritmos de factorización de matrices, como es el caso de la factorización de Cholesky, o la descomposición en valores singulares. Por último, estos resultados nos mostrarán que el cono de las matrices semidefinidas positivas permite generalizar algunos conceptos adicionales definidos para los números no negativos, como es el caso de las raíces cuadradas o el valor absoluto, manteniendo algunas de sus propiedades.

Comenzaremos con una caracterización de las matrices semidefinidas positivas por descomposición, la cual nos permitirá introducir además el concepto de raíz cuadrada. Utilizaremos para ello un lema previo.

**Lema 2.3.4.** *Sean $A, B \in \mathcal{M}_d(\mathbb{R})$ dos matrices que conmutan, es decir, $AB = BA$. Entonces, $Ap(B) = p(B)A$, donde $p$ denota cualquier polinomio sobre matrices (es decir, una expresión de la forma $p(C) = a_0I + a_1C + a_2C^2 + \dots a_nC^n$, con $a_1, \dots, a_n \in \mathbb{R}$).*
*Demostración.* Basta ver que

$$AB^n = (AB)B^{n-1} = B(AB)B^{n-2} = \dots = B^{n-1}(AB) = B^nA,$$

y $Ap(B) = p(B)A$ se deduce por linealidad. □

**Lema 2.3.5.** *Sea $D \in \mathcal{M}_d(\mathbb{R})_0^+$ una matriz diagonal. Entonces, existe un polinomio sobre matrices $p$ tal que $p(D^2) = D$.*
*Demostración.* Supongamos $D = \text{diag}(\lambda_1, \dots, \lambda_d)$, con $0 \leq \lambda_1 \leq \dots \lambda_d$. Entonces, $D^2 = \text{diag}(\lambda_1^2, \dots \lambda_d^2)$. Tomamos un polinomio $p$ de interpolación en los puntos $(\lambda_i^2, \lambda_i)$, para $i = 1, \dots, d$. Siempre podemos construir un polinomio de esta forma: si todos los $\lambda_i$ son distintos existe un único polinomio de grado $d-1$ que realiza esta interpolación (la matriz de Vandermonde asociada a los valores $\lambda_i^2$ es regular y los coeficientes del polinomio son la solución del sistema determinado por la matriz de Vandermonde y los valores $\lambda_i$ como término independiente) . Si alguno de los $\lambda_i$ está repetido, consideramos el punto $(\lambda_i^2, \lambda_i)$ una única vez, pudiendo interpolar mediante un polinomio de menor dimensión. El polinomio que obtenemos podemos evaluarlo sobre $D^2$, obteniendo

$$p(D^2) = p(\text{diag}(\lambda_1^2, \dots, \lambda_d^2)) = \text{diag}(p(\lambda_1^2), \dots, p(\lambda_d^2)) = \text{diag}(\lambda_1, \dots, \lambda_d) = D.$$

□

**Teorema 2.3.6.** *Sea $M \in \mathcal{M}_d(\mathbb{R})$. Entonces,*

a) *$M \in \mathcal{M}_d(\mathbb{R})_0^+$ si y solo si existe $L \in \mathcal{M}_d(\mathbb{R})$ tal que $M = L^T L$.*

b) *Si $M \in \mathcal{M}_d(\mathbb{R})_0^+$, existe una única matriz $N \in \mathcal{M}_d(\mathbb{R})_0^+$ tal que $N^2 = M$. Además, $M \in \mathcal{M}_d(\mathbb{R})^+ \iff N \in \mathcal{M}_d(\mathbb{R})^+$.*

*Demostración.* En primer lugar veamos que $L^T L$ es una matriz semidefinida positiva, para cualquier $L \in \mathcal{M}_d(\mathbb{R})$. En efecto, dado $x \in \mathbb{R}^d$,

$$x^T L^T L x = (Lx)^T (Lx) = \|Lx\|_2^2 \geq 0.$$

Probaremos la segunda implicación del primer apartado viendo directamente la existencia de la matriz $N$ del segundo apartado. Para ello, consideramos la descomposición $M = UDU^T$, con $U \in O_d(\mathbb{R})$ y $D = \operatorname{diag}(\lambda_1, \ldots, \lambda_d)$, con $0 \leq \lambda_1 \ldots \lambda_d$ los valores propios de $M$. Definimos $D^{1/2} = \operatorname{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_d})$ y construimos la matriz $N = UD^{1/2}U^T$. Se tiene que $N$ es semidefinida positiva, pues sus valores propios son los de $D^{1/2}$, que son todos positivos, y además,

$$N^2 = UD^{1/2}U^T UD^{1/2}U^T = UD^{1/2}D^{1/2}U^T = UDU^T = M.$$

También, la positividad estricta de los valores propios de $M$ equivale a la de los valores propios de $N$, luego $M \in \mathcal{M}_d(\mathbb{R})^+ \iff N \in \mathcal{M}_d(\mathbb{R})^+$. Veamos finalmente que $N$ es única.

Supongamos que existen $N_1, N_2 \in \mathcal{M}_d(\mathbb{R})_0^+$ con $N_1^2 = M = N_2^2$. Notemos que $N_1$ y $N_2$ han de tener los mismos valores propios, pues tienen todos sus valores propios reales por ser simétricas, los cuadrados de sus valores propios son los valores propios de $M$, y son semidefinidas positivas, luego sus valores propios son necesariamente las raíces positivas de los valores propios de $M$. Por tanto, son semejantes a una misma matriz diagonal, es decir, existen $U, V \in O_d(\mathbb{R})$ tales que $N_1 = UDU^T$ y $N_2 = VDV^T$. De $N_1^2 = N_2^2$ obtenemos

$$UD^2U^T = VD^2V^T \implies V^T UD^2 = D^2V^T U,$$

luego para $W = V^T U \in O_d(\mathbb{R})$, se tiene que $D^2$ y $W$ conmutan. Combinando los lemas 2.3.5 y 2.3.4 obtenemos que $D$ y $W$ también conmutan. Por tanto,

$$WD = DW \implies V^T UD = DV^T U \implies UDU^T = VDV^T \implies N_1 = N_2,$$

obteniéndose la unicidad. □

Como habíamos anticipado, este teorema motiva la definición de las raíces cuadradas para matrices semidefinidas positivas.

**Definición 2.3.1.** Sea $M \in \mathcal{M}_d(\mathbb{R})_0^+$. Se define la *raíz cuadrada* de $M$ como la única matriz $N \in \mathcal{M}_d(\mathbb{R})_0^+$ tal que $N^2 = M$. Dicha matriz además puede construirse como $N = UD^{1/2}U^T$, donde $M = UDU^T$ es una descomposición espectral de $M$. La raíz cuadrada se nota como $N = M^{1/2}$.

Observemos que una matriz, incluso siendo semidefinida positiva, puede admitir más de una matriz que tenga como cuadrado la matriz inicial. Por ejemplo, la identidad en $\mathbb{R}^2$ es el cuadrado de ella misma, de la simetría respecto al origen, o de la aplicación que intercambia los ejes de coordenadas. Lo que afirma el teorema es que, como ocurre con los números reales, la raíz cuadrada semidefinida positiva es única. La raíz cuadrada, además de seguir extendiendo las propiedades de los números no negativos al cono de matrices semidefinidas, es una herramienta útil para probar algunos resultados. Por ejemplo, aunque trabajemos con matrices simétricas, su producto no es necesariamente simétrico, luego podría no ser

diagonalizable por semejanza. Veamos que si una de las matrices admite raíces regulares podemos asegurar dicha diagonalización.

**Corolario 2.3.7.** *Si $A \in \mathcal{M}_d(\mathbb{R})^+$ y $B \in S_d(\mathbb{R})$, entonces $AB$ es diagonalizable por semejanza.*

*Demostración.* Se tiene que $A^{1/2} \in \mathcal{M}_d(\mathbb{R})^+$, y $A^{-1/2}(AB)A^{1/2} = A^{1/2}BA^{1/2}$. Esta última matriz es simétrica, pues $A^{1/2}$ y $B$ lo son, luego $(A^{1/2}BA^{1/2})^T = (A^{1/2})^T B^T (A^{1/2})^T = A^{1/2}BA^{1/2}$. El lado izquierdo de la igualdad nos dice que $AB$ es semejante a esta matriz, y por tanto, es diagonalizable. □

Continuando la extensión de conceptos sobre los números no negativos a matrices semidefinidas positivas, la raíz cuadrada permite definir el valor absoluto o módulo para matrices cuadradas arbitrarias.

**Definición 2.3.2.** Sea $A \in \mathcal{M}_d(\mathbb{R})$. Se define el *valor absoluto* o *módulo* de $A$ como

$$|A| = (A^T A)^{1/2} \in \mathcal{M}_d(\mathbb{R})_0^+.$$

Se denominan *valores singulares* de $A$ a los valores propios de $|A|$. Observemos que los valores singulares de cualquier matriz cuadrada son reales no negativos.

**Comentario 2.3.8.** *Si $A$ es simétrica y $A = UDU^T$ es una descomposición espectral, entonces los valores singulares de $A$ son el valor absoluto de sus valores propios y $|A| = U|D|U^T$.*

Como se ve en la definición, el módulo de matrices es siempre semidefinido positivo y su definición es análoga a las definiciones de módulo que pueden realizarse sobre $\mathbb{R}$ o $\mathbb{C}$. No todas las propiedades del valor absoluto pueden ser trasladadas al módulo de matrices. Por ejemplo, la desigualdad triangular (para el orden de Löwner) no se verifica. Sin embargo, sí es posible probar que para matrices $A, B \in \mathcal{M}_d(\mathbb{R})$ existen matrices ortogonales $U$ y $V$ tales que $|A+B| \preceq U|A|U^T + V|B|V^T$. Esta desigualdad se conoce como desigualdad de Thompson [33].

El módulo y los valores singulares permiten deducir nuevos teoremas de descomposición para matrices cuadradas arbitrarias.

**Teorema 2.3.9** (Descomposición polar y descomposición en valores singulares)**.** *Sea $A \in \mathcal{M}_d(\mathbb{R})$. Entonces,*

a) *Para todo $x \in \mathbb{R}^d$, $\|Ax\|_2 = \||A|x\|_2$.*

b) *Existe $U \in O_d(\mathbb{R})$ tal que $A = U|A|$. Esta descomposición se denomina* descomposición polar *de $A$, y no es necesariamente única.*

c) *Existen $V, W \in O_d(\mathbb{R})$ tales que $A = W\Sigma V^T$, donde $\Sigma$ es la matriz diagonal con los valores singulares de $A$. Esta descomposición se denomina* descomposición en valores singulares *de $A$.*

d) *Las columnas de $V$ forman una base ortonormal de vectores propios de $|A|$ (y de $A^TA$) y las columnas de $W$ forman una base ortonormal de vectores propios de $|A^T|$ (y de $AA^T$).*

*Demostración.*

a) Dado $x \in \mathbb{R}^d$,

$$\|Ax\|_2^2 = (Ax)^T(Ax) = x^T A^T Ax = x^T|A|^2 x = x^T|A||A|x = x^T|A|^T|A|x = (|A|x)^T(|A|x) = \||A|x\|_2^2.$$

b) Definimos la aplicación $U_1 \colon \operatorname{im}(|A|) \to \operatorname{im}(A)$ por $U_1(|A|x) = Ax$. Esta aplicación está bien definida, ya que $|A|x = |A|y \iff x - y \in \ker|A|$, y se tiene que $\ker|A| = \ker A$, ya que $|A|x = 0 \iff \||A|x\|_2 = \|Ax\|_2 = 0 \iff Ax = 0$. Por tanto, $|A|x = |A|y \iff x - y \in \ker A \iff Ax = Ay$.

Además, es una isometría, por el apartado anterior, y claramente es sobreyectiva. Se tiene además que

$$\dim \operatorname{im}(A)^\perp = d - \dim \operatorname{im}(A) = \dim \ker(A) = \dim \ker(|A|) = d - \dim \operatorname{im}(|A|) = \dim \operatorname{im}(|A|)^\perp.$$

Luego si $r = \dim \operatorname{im}(A)^\perp = \dim \operatorname{im}(|A|)^\perp$ podemos fijar bases ortonormales $\{u_1, \dots, u_r\}$ en $\operatorname{im}(A)^\perp$ y $\{w_1, \dots, w_r\}$ en $\operatorname{im}(|A|)^\perp$, y definir la aplicación lineal

$$\begin{aligned} U \colon \mathbb{R}^d &\to \mathbb{R}^d \\ x &\mapsto U_1 x, \quad x \in \operatorname{im}(|A|) \\ w_i &\mapsto u_i, \quad i = 1, \dots, r. \end{aligned}$$

Observemos que esta aplicación está bien definida, y por construcción es una isometría en $\mathbb{R}^d$. Por tanto, $U \in O_d(\mathbb{R})$ y $U|A| = U_1|A| = A$.

c) Consideramos la descomposición $|A| = V\Sigma V^T$, con $V \in O_d(\mathbb{R})$, y llamamos $W = UV \in O_d(\mathbb{R})$, donde $U$ es una matriz de descomposición polar dada por el apartado anterior. Entonces,

$$A = U|A| = UV\Sigma V^T = W\Sigma V^T.$$

d) La construcción de $V$ es clara, por la descomposición espectral $|A| = V\Sigma V^T$. Para $W$, notemos que

$$\Sigma^2 = V^T|A|^2V = V^TU^TAU^TAV = W^TAA^TUV = W^TAA^TW,$$

luego $W$ diagonaliza a $AA^T$ y por tanto también a $(AA^T)^{1/2} = |A^T|$.

□

**Comentario 2.3.10.** *Si $A$ es regular, $|A|$ también lo es, y la descomposición polar es única, donde la matriz $U$ viene dada por $U = A|A|^{-1}$.*

**Comentario 2.3.11.** *El módulo se puede definir de la misma forma para matrices no cuadradas $A \in \mathcal{M}_{d\times d'}(\mathbb{R})$. En tal caso, $|A| \in \mathcal{M}_{d'}(\mathbb{R})_0^+$. También es posible probar de forma análoga la existencia de descomposición polar y valores singulares. Para el caso de los valores singulares, las matrices $V$ y $W$ serán ortogonales con la dimensión adecuada para poder ser multiplicadas. En el caso de la descomposición polar, se tiene que $A = U|A|$, donde $U \in \lceil \times \lceil'(\mathbb{R})$ verifica $U^TU = I$, siempre que $d' \leq d$.*

La descomposición polar tiene una interpretación geométrica muy interesante, y es que nos dice que todo endomorfismo lineal en $\mathbb{R}^d$ es, salvo una isometría, un escalado sobre una determinada base ortonormal (la base donde diagonaliza $|A|$), entendiendo por escalado sobre dicha base una aplicación que a cada vector de la base (eje) lo envía a otro vector con la misma dirección y sentido (incluyendo el 0), pudiendo variar la constante de escala en cada eje.

La descomposición en valores singulares es una herramienta muy útil para cálculos con matrices por ordenador, como el cálculo de rangos, la resolución de sistemas de ecuaciones lineales, el cálculo de valores propios o los ajustes por mínimos cuadrados. Esto se debe a que se conocen algoritmos para calcularlos de forma eficiente, y permiten establecer niveles de tolerancia (por ejemplo, a la hora de determinar un rango) que hacen los cálculos más robustos frente a errores de precisión.

Para concluir con los teoremas de descomposición, vamos a afinar el resultado inicial con el que comenzábamos la sección (el teorema 2.3.6) añadiendo condiciones de unicidad sobre la descomposición $L^TL$. Para ello haremos uso de la descomposición polar.

**Teorema 2.3.12.** *Sea $M \in \mathcal{M}_d(\mathbb{R})_0^+$. Entonces,*

a) *Existe una matriz $L \in \mathcal{M}_d(\mathbb{R})$ tal que $M = L^T L$.*

b) *Si $K \in \mathcal{M}_d(\mathbb{R})$ es cualquier otra matriz tal que $M = K^T K$, entonces $K = UL$, donde $U \in O_d(\mathbb{R})$ (es decir, $L$ es única salvo isometrías).*

*Demostración.* La primera afirmación fue vista en el teorema 2.3.6. Supongamos entonces que $L, K \in \mathcal{M}_d(\mathbb{R})$ verifican $M = L^T L = K^T K$. Sean $L = V|L|, K = W|K|$, con $L, K \in O_d(\mathbb{R})$, descomposiciones polares de $L$ y $K$. Entonces,

$$\begin{aligned} L^T L = K^T K &\implies |L|^T V^T V |L| = |K|^T W^T W |K| \\ &\implies |L|^T |L| = |K|^T |K| \implies |L|^2 = |K|^2. \end{aligned}$$

Como $|L|$ y $|K|$ son semidefinidas positivas, han de ser la única raíz cuadrada de $|L|^2 = |K|^2$, es decir, $|L| = |K|$. Llamamos $N = |L| = |K|$. Volviendo a las descomposiciones polares de $L$ y $K$, se deduce que

$$N = V^T L = W^T K \implies K = W V^T L.$$

Por tanto, tomando $U = WV^T \in O_d(\mathbb{R})$ obtenemos la igualdad buscada. □

### 2.3.3. Matrices semidefinidas como seminormas

Es conocido que las matrices definidas positivas se identifican con productos escalares en $\mathbb{R}^d$, es decir, formas bilineales simétricas y definidas positivas. Para los productos escalares se tiene la conocida desigualdad de Cauchy-Schwarz, de la cual se deduce la desigualdad de Minkowski o desigualdad triangular, la cual permite concluir que la aplicación $\|\cdot\|_M \colon \mathbb{R}^d \to \mathbb{R}_0^+$ dada por $\|x\|_M = \sqrt{x^T M x}$ es una norma, para $M$ definida positiva. Cuando $M$ es únicamente semidefinida positiva no podemos asegurar que $\|\cdot\|_M$ sea una norma, pues en general no se tiene $\|x\| = 0 \iff x = 0$. Sin embargo, sí podemos probar, con algo más de esfuerzo, una desigualdad de Cauchy-Schwarz y una desigualdad triangular, que dan a $\|\cdot\|_M$ la condición de seminorma.

**Teorema 2.3.13** (Desigualdad de Cauchy-Schwarz para matrices semidefinidas positivas)**.** *Sea $M \in \mathcal{M}_d(\mathbb{R})_0^+$ y definimos $g \colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ por $g(x, y) = x^T M y$. Entonces,*

$$|g(x, y)| \leq \|x\|_M \|y\|_M.$$

*Demostración.* Supongamos que $g(y, y) = 0$, y sea $\lambda \in \mathbb{R}^+$ arbitrario. Entonces,

$$0 \leq g(x + \lambda y, x + \lambda y) = g(x, x) + 2\lambda g(x, y) + \lambda^2 g(y, y) = g(x, x) + 2\lambda g(x, y).$$

Por tanto,

$$-\frac{g(x, x)}{2\lambda} \leq g(x, y).$$

Razonando análogamente para $-\lambda$, se obtiene que

$$-\frac{g(x, x)}{2\lambda} \leq g(x, y) \leq \frac{g(x, x)}{2\lambda}.$$

La arbitrariedad de $\lambda$ conduce a que $g(x, y) = 0$, verificándose la desigualdad.

Supongamos ahora $g(y,y) \neq 0$. Análogamente a la desigualdad de Cauchy-Schwarz para el producto escalar, consideramos $\lambda = \frac{g(x,y)}{g(y,y)}$. Entonces,

$$\begin{aligned} 0 \leq g(x-\lambda y, x-\lambda y) &= g(x,x) - 2\lambda g(x,y) + \lambda^2 g(y,y) \\ &= g(x,x) - 2\frac{g(x,y)^2}{g(y,y)} + \frac{g(x,y)^2}{g(y,y)} = \|x\|_M^2 - \frac{g(x,y)^2}{\|y\|_M^2}, \end{aligned}$$

de donde se deduce que

$$g(x,y)^2 \leq \|x\|_M^2 \|y\|_M^2.$$

□

**Corolario 2.3.14** (Desigualdad triangular o de Minkowski)**.** *Si $M \in \mathcal{M}_d(\mathbb{R})_0^+$, entonces $\|x+y\|_M \leq \|x\|_M + \|y\|_M$.*

*Demostración.* Aplicando la desigualdad de Cauchy-Schwarz,

$$\|x+y\|_M^2 = \|x\|_M^2 + 2g(x,y) + \|y\|_M^2 \leq \|x\|_M^2 + 2\|x\|_M\|y\|_M + \|y\|_M^2 = (\|x\|_M + \|y\|_M)^2$$

□

### 2.3.4. Proyección sobre las matrices semidefinidas

Para concluir la sección, retomamos la estructura del conjunto de matrices semidefinidas como cono convexo y cerrado. El teorema de la proyección convexa 1.1.4 nos dice toda matrix podemos proyectarla a dicho cono. Veremos que podemos calcular una proyección explícita.

Antes de analizar la proyección sobre el cono de matrices semidefinidas, analizamos otra proyección importante desde el punto de vista algorítmico: la proyección sobre el espacio vectorial de las matrices simétricas. Es fácil comprobar que $S_d(\mathbb{R}) \perp A_d(\mathbb{R})$, para el producto escalar de Frobenius, y además $S_d(\mathbb{R}) \bigoplus A_d(\mathbb{R}) = \mathcal{M}_d(\mathbb{R})$, pues su intersección es vacía, al ser ortogonales, y sus dimensiones suman $d^2$ (el espacio de las matrices simétricas tiene dimensión $d(d+1)/2$ y el de las antisimétricas $d(d-1)/2$; esto se debe a que las matrices simétricas vienen determinadas por sus componentes en el triángulo superior con la diagonal, y las antisimétricas vienen determinadas por sus componentes en el triángulo superior sin la diagonal). Por tanto, $S_d(\mathbb{R})$ y $A_d(\mathbb{R})$ se complementan ortogonalmente. Observando que, para toda $A \in \mathcal{M}_d(\mathbb{R})$, la descomposición de $A$ en estos subespacios es

$$A = \frac{A + A^T}{2} + \frac{A - A^T}{2},$$

donde el primer sumando es simétrico y el segundo antisimétrico, el teorema de la proyección ortogonal permite concluir que la proyección sobre $S_d(\mathbb{R})$ viene dada por $A \mapsto (A + A^T)/2$.

Pasamos a buscar la proyección sobre el cono de las matrices semidefinidas. Veremos en primer lugar que, cuando queremos proyectar matrices simétricas, la proyección tiene una expresión muy sencilla, a partir de los valores propios.

**Definición 2.3.3.** Sea $\Sigma \in \mathcal{M}_d(\mathbb{R})$ una matriz diagonal, $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_d)$. Se define la *parte positiva* de $\Sigma$ como $\Sigma^+ = \operatorname{diag}(\sigma_1^+, \ldots, \sigma_d^+)$, donde $\sigma_i^+ = \max\{\sigma_i, 0\}$. Análogamente, se define su *parte negativa* como $\Sigma^- = \operatorname{diag}(\sigma_1^-, \ldots, \sigma_d^-)$, donde $\sigma_i = \max\{-\sigma_i, 0\}$.

Sea $A \in S_d(\mathbb{R})$ y sea $A = UDU^T$ una descomposición espectral. Se define la *parte positiva* de $A$ como $A^+ = UD^+U^T$. Análogamente, se define la parte negativa de $A$ como $A^- = UD^-U^T$.

Es fácil comprobar que $A^+$ no depende de la matriz $U$ escogida, como consecuencia del lema 2.3.4 aplicado a un polinomio que interpole los puntos de la forma $(\lambda_i, \lambda_i^+)$ (el teorema 2.3.6 muestra el procedimiento). Lo mismo ocurre con $A^-$. Además, observemos que $A^+, A^- \in \mathcal{M}_d(\mathbb{R})_0^+$. $A^+$ determinará la proyección de matrices simétricas sobre el cono de matrices semidefinidas positivas.

**Teorema 2.3.15** (Proyección semidefinida)**.** *Sea $A \in S_d(\mathbb{R})$. Entonces, $A^+$ es la proyección de $A$ sobre el cono de las matrices semidefinidas positivas.*

*Demostración.* Tomamos $A = UDU^T$, con $U \in O_d(\mathbb{R})$, una descomposición espectral de $A$, con $D = \operatorname{diag}(\lambda_1, \dots, \lambda_d)$, y $\lambda_i \in \mathbb{R}$ los valores propios de $A$, para $i = 1, \dots, d$. Sea $M \in \mathcal{M}_d(\mathbb{R})_0^+$ arbitraria. Llamamos $S = U^T M U$ (esto es, $M = USU^T$). Se tiene que

$$\begin{aligned}\|A - M\|_F^2 = \|UDU^T - USU^T\|_F^2 = \|U(D-S)U^T\|_F^2 = \|D - S\|_F^2 \\ = \sum_{i \neq j} S_{ij}^2 + \sum_{i=1}^d (\lambda_i - S_{ii})^2 \geq \sum_{i=1}^d (\lambda_i - S_{ii})^2 \\ \geq \sum_{\lambda_i < 0} (\lambda_i - S_{ii})^2 \geq \sum_{\lambda_i < 0} \lambda_i^2,\end{aligned}$$

donde en la última desigualdad se ha usado que $S_{ii} \geq 0$ para cada $i \in \{1, \dots, d\}$, por ser $S$ semidefinida positiva (basta ver que $S_{ii} = e_i^T S e_i \geq 0$, donde $\{e_1, \dots, e_d\}$ es la base canónica de $\mathbb{R}^d$), y por tanto, $(\lambda_i - S_{ii})^2 = \lambda_i^2 - 2\lambda_i S_{ii} + S_{ii}^2 \geq \lambda_i^2$ para cada $i$ con $\lambda_i < 0$.

La desigualdad anterior es válida para toda $M \in \mathcal{M}_d(\mathbb{R})_0^+$, y solo depende de $A$ (concretamente, de sus valores propios negativos). Notemos que para $A^+$ se da la igualdad, pues

$$\|A - A^+\|_F^2 = \|D - D^+\|_F^2 = \sum_{\lambda_i < 0} \lambda_i^2,$$

luego $A^+$ minimiza la distancia a $A$ dentro de $\mathcal{M}_d(\mathbb{R})_0^+$.

El teorema de la proyección convexa 1.1.4 asegura que $A^+$ es la proyección de $A$ sobre el cono de las matrices semidefinidas positivas, aunque también es fácil comprobarlo con la desigualdad anterior, pues cuando se da la igualdad, se ha de verificar que $S_{ij} = 0$ para $i \neq j$, $S_{ii} = 0$ para $\lambda_i < 0$ y $S_{ii} = \lambda_i$ en caso contrario. □

Concluimos viendo la proyección semidefinida de una matriz cuadrada arbitraria.

**Corolario 2.3.16.** *Sea $A \in \mathcal{M}_d(\mathbb{R})$. Entonces, la proyección de $A$ sobre el cono de las matrices semidefinidas positivas es $((A + A^T)/2)^+$.*

*Demostración.* Podemos descomponer $A = B + C$, donde $B = (A + A^T)/2 \in S_d(\mathbb{R})$ y $C = (A - A^T)/2 \in A_d(\mathbb{R})$. Como $\langle B, C \rangle_F = 0$, el teorema de Pitágoras nos dice que, para $M \in \mathcal{M}_d(\mathbb{R})_0^+$, que también verifica $\langle M, C \rangle_F = 0$, se tiene

$$\|A - M\|_F^2 = \|B - M\|_F^2 + \|C\|_F^2.$$

Por tanto, minimizar la distancia a $A$ en $\mathcal{M}_d(\mathbb{R})_0^+$ equivale a minimizar la distancia a $B$ en el mismo conjunto, la cual alcanza el mínimo cuando $M = B^+$. □

**Comentario 2.3.17.** *Los teoremas de proyección semidefinida nos permiten calcular también la distancia de una matriz al cono de matrices semidefinidas positivas. Si $A \in \mathcal{M}_d(\mathbb{R})$, y $B$ y $C$ son sus partes simétrica y antisimétrica, con $\lambda_i(B)$ los valores propios de $B$, para $i = 1, \dots, d$, esta distancia viene dada por*

$$d(A, \mathcal{M}_d(\mathbb{R})_0^+) = \sum_{\lambda_i(B) < 0} \lambda_i(B)^2 + \|C\|_F^2.$$

**Comentario 2.3.18.** *Análogamente se tiene que la proyección sobre el cono de matrices semidefinidas negativas de una matriz simétrica $A \in \mathcal{M}_d(\mathbb{R})$ es $-A^-$. Las partes positiva y negativa, como ocurre con los números reales, permiten recuperar la matriz original, es decir, se verifica que $A = A^+ - A^-$. Igualmente ocurre con el módulo: $|A| = A^+A^-$. Esto permite calcular la proyección semidefinida también como $A^+ = (A + |A|)/2$.*

## 2.4. Cociente de Rayleigh. Optimización con vectores propios.

En esta sección estudiaremos varios problemas de optimización matricial cuya resolución involucra valores y vectorios propios de las matrices que definen el problema. La resolución de estos problemas nos será de gran utilidad en los algoritmos que estudiaremos más adelante.

**Definición 2.4.1.** Sea $A \in S_d(\mathbb{R})$. Se define el *cociente de Rayleigh* asociado a $A$ como la aplicación $\rho_A \colon \mathbb{R}^d \setminus \{0\} \to \mathbb{R}$ dada por

$$\rho_A(x) = \frac{x^T Ax}{x^T x} = \frac{\langle Ax, x\rangle}{\|x\|_2^2} \quad \forall x \in \mathbb{R}^d \setminus \{0\}.$$

Si $B \in \mathcal{M}_d(\mathbb{R})^+$, se define el *cociente de Rayleigh generalizado* asociado a $A$ y $B$ como la aplicación $\mathcal{R}_{A,B} \colon \mathbb{R}^d \setminus \{0\} \to \mathbb{R}$ dada por

$$\mathcal{R}_{A,B}(x) = \frac{x^T Ax}{x^T Bx} = \frac{\langle Ax, x\rangle}{\|x\|_B^2} \quad \forall x \in \mathbb{R}^d \setminus \{0\}.$$

Ambas aplicaciones están bien definidas, pues los cocientes solo pueden anularse cuando $x = 0$. Notemos que el cociente de Rayleigh es un caso particular del cociente generalizado cuando $B = I$. A lo largo de la sección supondremos $A \in S_d(\mathbb{R})$ y $B \in \mathcal{M}_d(\mathbb{R})^+$ fijas, y nos referiremos a los cocientes de Rayleigh como $\rho = \rho_A$ y $\mathcal{R} = \mathcal{R}_{a,b}$.

Una primera observación sobre estas aplicaciones es que, para $x \in \mathbb{R}^d \setminus \{0\}$ y $\lambda \in \mathbb{R}^*$, se verifica que

$$\mathcal{R}(\lambda x) = \frac{(\lambda x)^T A(\lambda x)}{(\lambda x)^T B(\lambda x)} = \frac{\lambda^2 (x^T Ax)}{\lambda^2 (x^T Bx)} = \mathcal{R}(x).$$

Por tanto, $\mathcal{R}$ toma todos sus valores en la esfera unidad $d-1$-dimensional, es decir $\mathcal{R}(\mathbb{R} \setminus \{0\}) = \mathcal{R}(\mathbb{S}^{d-1}) \subset \mathbb{R}$. Como $\mathcal{R}$ es continua y la esfera $d-1$-dimensional es compacta, se tiene que $\mathcal{R}$ alcanza un máximo y un mínimo en $\mathbb{R}^d \setminus \{0\}$, los cuales no van a ser únicos. Igualmente ocurre con $\rho$.

Podemos encontrar los valores máximos y mínimos que toman los cocientes de Rayleigh. Comenzaremos analizando $\rho$.

**Teorema 2.4.1** (Rayleigh-Ritz)**.** *Sean $\lambda_{\text{mín}}$ y $\lambda_{\text{máx}}$ los valores propios mínimo y máximo, respectivamente, de $A$. Entonces,*

a) *Para todo $x \in \mathbb{R}^d$, se tiene que $\lambda_{\text{mín}}\|x\|^2 \leq x^T Ax \leq \lambda_{\text{máx}}\|x\|^2$.*

b) $\lambda_{\text{máx}} = \text{máx}_{x \in \mathbb{R}^d \setminus \{0\}} \frac{x^T Ax}{x^T x} = \text{máx}_{\|x\|_2 = 1} x^T Ax$.

c) $\lambda_{\text{mín}} = \text{mín}_{x \in \mathbb{R}^d \setminus \{0\}} \frac{x^T Ax}{x^T x} = \text{mín}_{\|x\|_2 = 1} x^T Ax$.

*Por tanto, los valores máximo y mínimo de $\rho$ son $\lambda_{\text{máx}}$ y $\lambda_{\text{mín}}$, respectivamente.*

*Demostración.* Sea $A = UDU^T$ con $U \in O_d(\mathbb{R})$ y $D = \operatorname{diag}(\lambda_1, \dots, \lambda_d)$, donde $\lambda_1 \leq \cdots \leq \lambda_d$, una descomposición espectral de $A$. Sea $x \in \mathbb{R}^d \setminus \{0\}$ y tomamos $y = U^T x$. Entonces,

$$\rho(x) = \frac{x^T A x}{x^T x} = \frac{x^T U D U^T x}{x^T x} = \frac{y^T U^T U D U^T U y}{y^T U^T U y} = \frac{y^T D y}{\|y\|_2^2} = \frac{\sum_{i=1}^{d} \lambda_i y_i^2}{\|\lambda\|_2^2}. \tag{2.1}$$

Además, es claro que

$$\lambda_1 \|y\|_2^2 = \lambda_1 \sum_{i=1}^{d} y_i^2 \leq \sum_{i=1}^{d} \lambda_i y_i^2 \leq \lambda_d \sum_{i=1}^{d} y_i^2 = \lambda_d \|y\|_2^2.$$

Aplicando esta desigualdad sobre la expresión 2.1, se deduce que

$$\lambda_1 \leq \rho(x) \leq \lambda_d.$$

Además, si $u_1$ y $u_d$ son vectores propios de $A$ asociados a $\lambda_1$ y $\lambda_d$, se tiene que

$$\rho(u_1) = \frac{u_1^T A u_1}{u_1^T u_1} = \frac{\lambda_1 u_1^T u_1}{u_1^T u_1} = \lambda_1, \quad \rho(u_d) = \frac{u_d^T A u_d}{u_d^T u_d} = \frac{\lambda_d u_d^T u_d}{u_d^T u_d} = \lambda_d.$$

Por tanto, se alcanza la igualdad, lo que permite deducir las tres afirmaciones del teorema. □

El teorema de Rayleigh-Ritz nos muestra que $\rho(\mathbb{R}^d \setminus \{0\}) = [\lambda_{\text{mín}}, \lambda_{\text{máx}}]$, alcanzándose un mínimo o máximo en los respectivos vectores propios. Sin embargo, $\lambda_{\text{mín}}$ y $\lambda_{\text{máx}}$ no son los únicos valores propios que actúan como óptimos para el cociente de Rayleigh. Si nos restringimos a espacios de menor dimensión, podemos obtener cualquier valor propio de $A$ como óptimo para el cociente de Rayleigh, como veremos en los siguientes resultados.

**Lema 2.4.2.** *Sean $\lambda_1, \dots, \lambda_d$ los valores propios de $A$, y $u_1, \dots, u_d$ vectores propios ortonormales asociados. Entonces, para cada $k \in \{1 \dots, d\}$,*

$$\lambda_k = \max_{x \in \operatorname{lin}\{u_1, \dots, u_k\}} \rho(x) = \min_{x \in \operatorname{lin}\{u_k, \dots, u_d\}} \rho(x) \tag{2.2}$$

*Demostración.* Si $x \in \operatorname{lin}\{u_1, \dots, u_{d-k+1}\}$, entonces $x = \sum_{i=1}^{d-k+1} \langle x, u_i \rangle u_i$, y por la identidad de Parseval,

$$x^T x = \langle x, x \rangle = \sum_{i=1}^{d-k+1} \langle x, u_i \rangle^2$$
$$x^T A x = \langle x, Ax \rangle = \sum_{i=1}^{d-k+1} \lambda_i \langle x, u_i \rangle^2.$$

Si tomamos $y_i = \langle x, u_i \rangle$, tenemos una expresión análoga a la de 2.1, que en este caso tiene como valor máximo $\lambda_k$, y se alcanza en $u_k$. Razonando análogamente sobre $\operatorname{lin}\{u_k, \dots, u_d\}$, obtenemos que el cociente de Rayleigh en este subespacio tiene como mínimo $\lambda_k$ y se alcanza en $u_k$. □

**Teorema 2.4.3** (Courant-Fischer)**.** *Sean $\lambda_1 \leq \cdots \leq \lambda_d$ los valores propios de $A$ y notamos por $S_k$ a un subespacio vectorial de $\mathbb{R}^d$ de dimensión $k$. Entonces, para cada $k \in \{1, \dots, d\}$, se tiene que*

$$\lambda_k = \min_{S_k \subset \mathbb{R}^d} \max_{\substack{x \in S_k \\ \|x\|_2 = 1}} x^T A x \tag{2.3}$$

$$\lambda_k = \max_{S_{d-k+1} \subset \mathbb{R}^d} \min_{\substack{x \in S_{d-k+1} \\ \|x\|_2 = 1}} x^T A x \tag{2.4}$$

*Demostración.* Probaremos que se verifica la igualdad 2.3. La igualdad 2.4 se prueba de forma análoga.

Sean $u_1, \dots, u_d$ vectores propios ortonormales asociados a $\lambda_1, \dots, \lambda_d$. El lema 2.4.2 nos dice que

$$\lambda_k = \max_{\substack{x \in \operatorname{lin}\{u_1,\dots,u_k\} \\ \|x\|_2 = 1}} x^T A x.$$

Como $\dim(\operatorname{lin}\{u_1, \dots, u_k\}) = k$, se tiene que

$$\min_{S_k \subset \mathbb{R}^d} \max_{\substack{x \in S_k \\ \|x\|_2 = 1}} x^T A x \leq \lambda_k.$$

Por otro lado, sea $A = UDU^T$ con $U \in O_d(\mathbb{R})$ y $D = \operatorname{diag}(\lambda_1, \dots, \lambda_d)$, donde $\lambda_1 \leq \cdots \leq \lambda_d$ la descomposición espectral de $A$ asociada a la base $\{u_1, \dots, u_d\}$. Fijado $S_k$ un subespacio de dimensión $k$, definimos los subespacios vectoriales

$$\begin{aligned} V &= \{U^T x \colon x \in S_k\} \\ W &= \{y \in \mathbb{R}^d \colon y_1 = \cdots = y_{k-1} = 0\} = \operatorname{lin}\{e_k, \dots, e_d\}, \end{aligned}$$

donde $\{e_1, \dots, e_d\}$ es la base canónica de $\mathbb{R}^d$. Como $\dim(V) = \dim(S_k) = k$ y $\dim(W) = d - k + 1$. Como $\dim(V) + \dim(W) = d + 1 > d$, necesariamente $\dim(V \cap W) \geq 1$. Por tanto, $V \cap W$ contiene vectores no nulos para cualquier $S_k$, y entonces,

$$\begin{aligned} \min_{S_k \subset \mathbb{R}^d} \max_{\substack{x \in S_k \\ \|x\|_2 = 1}} x^T A x &= \min_{V \subset \mathbb{R}^d} \max_{\substack{y \in V \\ \|y\|_2 = 1}} y^T D y \\ &\geq \min_{V \subset \mathbb{R}^d} \max_{\substack{y \in V \cap W \\ \|y\|_2 = 1}} y^T D y \\ &= \min_{V \subset \mathbb{R}^d} \max_{\substack{y \in V \cap W \\ \|y\|_2 = 1}} \sum_{i=k}^{d} \lambda_i y_i^2 \\ &\geq \lambda_k, \end{aligned}$$

obteniendo así la desigualdad restante. □

Los teoremas de Rayleigh-Ritz y de Courant-Fischer tienen consecuencias muy interesantes. Por ejemplo, una de ellas es el teorema de Weyl, que relaciona los valores propios de dos matrices simétricas con el de su suma. De este teorema se puede deducir que el orden de Löwner sobre las matrices es equivalente al orden producto sobre los vectores con los valores propios. Estos resultados no se usarán a lo largo de este trabajo, pero pueden consultarse en [19, 41]. Sí que utilizaremos el conocido como teorema de entrelace de Cauchy.

**Teorema 2.4.4** (Entrelace de Cauchy)**.** *Supongamos que $\lambda_1 \leq \cdots \leq \lambda_d$ son los valores propios de $A$. Sea $J \subset \{1, \dots, d\}$ de cardinal $|J| = d'$, y sea $A_J \in S_{d'}(\mathbb{R})$ la matriz $A_J = (A_{ij})_{i,j \in J}$, es decir, la submatriz de $A$ con las entradas de $A$ cuyos índices están en $J \times J$. Entonces, si $\tau_1 \leq \cdots \leq \tau_{d'}$ son los valores propios de $A_J$, se tiene que, para cada $k \in \{1, \dots, d'\}$,*

$$\lambda_k \leq \tau_k \leq \lambda_{k+d-d'}.$$

*Demostración.* Veamos la prueba para el caso en que $J = \{1, \dots, d'\}$. La prueba general es igual, eligiendo en cada caso un subespacio $W$ generado por vectores de la base canónica $e_1, \dots, e_d$ cuyos índices se adapten a los del conjunto $J$.

Sea $k \in \{1, \dots, d\}$. Consideramos el subespacio $W = \lin\{e_1, \dots, e_{d'}\} = \{x \in \mathbb{R}^d \colon e_{d'+1} = \cdots = e_d = 0\}$. Dado $x \in W$ denotaremos $\pi(x) = (x_1, \dots, x_{d'}) \in \mathbb{R}^{d'}$ al vector con las componentes no nulas de $x$ (o la correspondiente identificación sobre $\mathbb{R}^{d'}$). Observemos que $\pi(x)^T A_J \pi(x) = \sum_{i,j=1}^{d'} A_{ij} x_i x_j = x^T A x$. Aplicando el teorema de Courant-Fischer 2.4.3 restringiéndonos a subespacios de $W \equiv \mathbb{R}^{d'}$ de dimensión $k$, se tiene que

$$\lambda_k = \min_{S_k \subset \mathbb{R}^d} \max_{\substack{x \in S_k \\ \|x\|_2 = 1}} x^T A x \leq \min_{S_k \subset W} \max_{\substack{x \in S_k \\ \|x\|_2 = 1}} \pi(x)^T A_J \pi(x) = \tau_k$$

De la misma forma, si nos restringimos a subespacios de $W$ de dimensión $d' + 1 - k = d - (k + d - d') + 1$, obtenemos que

$$\lambda_{k+d-d'} = \max_{S_{d'+1-k} \subset \mathbb{R}^d} \min_{\substack{x \in S_{d'+1-k} \\ \|x\|_2 = 1}} x^T A x \geq \max_{S_{d'+1-k} \subset W \equiv \mathbb{R}^{d'}} \min_{\substack{x \in S_{d'+1-k} \\ \|x\|_2 = 1}} \pi(x)^T A_J \pi(x) = \tau_k$$

□

**Corolario 2.4.5.** *Sea $L \in \mathcal{M}_{d' \times d}(\mathbb{R})$ tal que $LL^T = I$. Si $\mu_1 \geq \cdots \geq \mu_d$ son los valores propios de $A$, y $\sigma_1 \geq \cdots \geq \sigma_{d'}$ son los valores propios de $LAL^T$ (en este caso estamos tomando los valores propios ordenados descendentemente), entonces, $\sigma_k \leq \mu_k$, para $k = 1, \dots, r$.*

*Demostración.* La condición $LL^T = I$ nos dice que $L$ es parcialmente una isometría de un subespacio $V$ de $\mathbb{R}^d$ de dimensión $d'$ sobre $\mathbb{R}^{d'}$. Si fijamos una base en $\mathbb{R}^{d'}$, la imagen de dicha base por $L^T$ es una base de $V$. Extendiendo a una base de $\mathbb{R}^d$, la matriz de $A$ (como aplicación lineal fijada la base usual) en esta nueva base es una matriz semejante a $A$. Llamémosla $A'$. Entonces, la matriz de la aplicación $LAL^T$ sobre la base escogida de $\mathbb{R}^{d'}$ es una submatriz de $A'$, y en consecuencia $LAL^T$ es submatriz de una matriz semejante a $A$, por lo que podemos aplicar el teorema de entrelace 2.4.4 sobre $A$ y $LAL^T$. La desigualdad buscada se deduce al reordenar de forma descendente los valores propios del enunciado. □

A continuación planteamos el problema de optimización que buscamos resolver en esta sección. Queremos, para cada $d' \leq d$, maximizar la suma de cocientes de Rayleigh sobre vectores ortogonales, esto es,

$$\max_{\substack{u_1, \dots, u_{d'} \in \mathbb{R}^d \\ u_i \neq 0 \\ \langle u_i, u_j \rangle = 0 (i \neq j)}} \sum_{i=1}^{d'} \rho(u_i). \tag{2.5}$$

Teniendo en cuenta que maximizar el cociente de Rayleigh equivale a hacerlo sobre vectores unitarios, y si colocamos los vectores $u_i$ por columnas en una matriz $L \in \mathcal{M}_{d' \times d}(\mathbb{R})$, que los vectores formen una conjunto ortonormal equivale a que $L^T L = I$, podemos reescribir el problema como

$$\max_{\substack{u_1, \dots, u_{d'} \in \mathbb{R}^d \\ u_i \neq 0 \\ \langle u_i, u_j \rangle = 0 (i \neq j)}} \sum_{i=1}^{d'} \rho(u_i) = \max_{\substack{u_1, \dots, u_{d'} \in \mathbb{R}^d \\ \|u_i\|_2 = 1 \\ \langle u_i, u_j \rangle = 0 (i \neq j)}} \sum_{i=1}^{d'} u_i^T A u_i = \max_{\substack{L \in \mathcal{M}_{d' \times d}(\mathbb{R}) \\ LL^T = I}} \operatorname{tr}(LAL^T). \tag{2.6}$$

Veamos cómo resolver este problema.

**Teorema 2.4.6** (Optimización de la traza por vectores propios)**.** *Sean $d', d \in \mathbb{N}$, con $d' \leq d$. Sea $A \in \mathcal{S}_d(\mathbb{R})$, y consideramos el problema de optimización*

$$\begin{aligned} \max_{L \in \mathcal{M}_{d' \times d}(\mathbb{R})} \quad & \operatorname{tr}\left(LAL^T\right) \\ \text{s.a.:} \quad & LL^T = I. \end{aligned} \tag{2.7}$$

*Entonces, el problema alcanza un máximo si* $L = \begin{pmatrix} — & v_1 & — \\ & \dots & \\ — & v_{d'} & — \end{pmatrix}$, *donde* $v_1, \dots, v_{d'}$ *son vectores propios ortonormales de* $A$ *correspondientes a sus* $d'$ *mayores valores propios. Además, el valor máximo es la suma de los* $d'$ *mayores valores propios de* $A$.

*Demostración.* Sean $\mu_1 \geq \cdots \geq \mu_d$ los valores propios de $A$, ordenados decrecientemente, y $\sigma_1 \geq \cdots \geq \sigma_{d'}$ los valores propios de $LAL^T$. Por el corolario 2.4.5, para cualquier $L \in \mathcal{M}_{d' \times d}(\mathbb{R})$ con $LL^T = I$,

$$\operatorname{tr}(LAL^T) = \sum_{i=1}^{d'} \sigma_i \leq \sum_{i=1}^{d'} \mu_i.$$

Además, cuando $L$ contiene, por filas, los vectores propios $v_1, \dots, v_{d'}$ de $A$, se tiene que $LL^T = I$ y $\operatorname{tr}(LAL^T) = \sum_{i=1}^{d'} \mu_i$, luego la cota anterior se alcanza justo en estos vectores. □

Finalmente, aplicamos los resultados ya probados al cociente de Rayleigh generalizado. Para ello, veamos que podemos diagonalizar (por congruencia) las matrices $A$ y $B$ simultáneamente.

**Lema 2.4.7** (Diagonalización simultánea de matrices)**.** *Sea* $A \in S_d(\mathbb{R})$ *y* $B \in \mathcal{M}_d(\mathbb{R})_0^+$. *Entonces existe una matriz regular* $P \in \mathrm{GL}_d(\mathbb{R})$ *y una matriz diagonal* $D \in \mathcal{M}_d(\mathbb{R})$ *tales que* $P^T AP = D$ *y* $P^T BP = I$.

*Demostración.* Consideramos la matriz $C = B^{-1/2}AB^{-1/2}$. $C$ es claramente simétrica, por serlo $A$ y $B$, luego existe $U \in O_d(\mathbb{R})$ tal que $U^T CU$ es diagonal. Llamamos $D = U^T CU$ y tomamos $P = B^{-1/2}U \in \mathrm{GL}_d(\mathbb{R})$. Se tiene que

$$P^T AP = P^T B^{1/2} C B^{1/2} P = (B^{-1/2}U)^T B^{1/2} C B^{1/2} (B^{-1/2}U) = U^T (B^{-1/2}B^{1/2}) C (B^{1/2}B^{-1/2}) U = D$$
$$P^T BP = (B^{-1/2}U)^T B (B^{-1/2}U) = U^T B^{-1/2} B B^{-1/2} U = U^T U = I.$$

□

El lema anterior nos permite expresar el cociente de Rayleigh generalizado a partir de un cociente de Rayleigh para la matriz $D$ diagonal anterior. En efecto, para $x \in \mathbb{R}^d \setminus \{0\}$, tomando la matriz regular $P$ del lema anterior, podemos tomar $y = P^{-1}x \in \mathbb{R}^d \setminus \{0\}$, y entonces

$$\mathcal{R}(x) = \frac{x^T Ax}{x^T Bx} = \frac{(Py)^T A(Py)}{(Py)^T B(Py)} = \frac{y^T P^T APy}{y^T P^T BPy} = \frac{y^T Dy}{y^T y} = \rho_D(y).$$

Por tanto, $\mathcal{R}(x) = \rho_D(P^{-1}x)$, y maximizar o minimizar $\mathcal{R}$ equivale a maximizar o minimizar $\rho_D$. Observemos que podemos suponer $D = \operatorname{diag}(\lambda_1, \dots, \lambda_d)$, con $\lambda_1 \geq \cdots \geq \lambda_d$ ($D$ contiene los valores propios de $B^{-1/2}AB^{-1/2}$, como se muestra en el lema previo). Entonces, la imagen de $\mathcal{R}$ está en el intervalo $[\lambda_d, \lambda_1]$, e igualmente se puede probar un teorema de Courant-Fischer asociado a $\mathcal{R}$.

De la misma forma, podemos concluir el siguiente resultado de optimización, que extiende al del problema del teorema 2.4.6.

**Teorema 2.4.8.** *Sean* $d', d \in \mathbb{N}$, *con* $d' \leq d$. *Sean* $A \in \mathcal{S}_d(\mathbb{R})$ *y* $B \in \mathcal{M}_d(\mathbb{R})^+$, *y consideramos el problema de optimización*

$$\max_{L \in \mathcal{M}_{d' \times d}(\mathbb{R})} \operatorname{tr}\left((LBL^T)^{-1}(LAL^T)\right) \tag{2.8}$$

*Entonces, el problema alcanza un máximo si* $L = \begin{pmatrix} — & v_1 & — \\ & \dots & \\ — & v_{d'} & — \end{pmatrix}$, *donde* $v_1, \dots, v_{d'}$ *son los vectores propios de* $B^{-1}A$ *correspondientes a sus* $d'$ *mayores valores propios.*

*Demostración.* Llamamos $U = L^T \in \mathcal{M}_{d\times d'}(\mathbb{R})$. Si tomamos la matriz $P$ del lema de diagonalización simultánea 2.4.7 y una matriz $V \in \mathcal{M}_{d\times d'}(\mathbb{R})$ tal que $U = PV$ (existe y es única por ser $P$ regular), se tiene

$$\begin{aligned}\operatorname{tr}\left((LBL^T)^{-1}(LAL^T)\right) &= \operatorname{tr}\left((U^TBU)^{-1}(U^TAU)\right) = \operatorname{tr}\left((V^TP^TBPV)^{-1}(V^TP^TAPV)\right)\\ &= \operatorname{tr}((V^TV)^{-1}(V^TDV)),\end{aligned}$$

luego maximizar la ecuación 2.8 equivale a maximizar en $V$ $\operatorname{tr}((V^TV)^{-1}(V^TDV))$, pues el cambio de parámetros entre $L$ y $V$ es biyectivo. Si ahora consideramos $V = Q|V|$ una descomposición polar de $V$ (generalizada, según la observación 2.3.11), donde $Q \in \mathcal{M}_{d\times d'}(\mathbb{R})$ verifica $Q^TQ = I$, se tiene que

$$\begin{aligned}\operatorname{tr}((V^TV)^{-1}(V^TDV)) &= \operatorname{tr}((|V|^TQ^TQ|V|)^{-1}(|V|^TQ^TDQ|V|^T)) = \operatorname{tr}(|V|^{-1}|V|^{-T}(|V|^TQ^TDQ|V|))\\ &= \operatorname{tr}(|V|^{-1}Q^TDQ|V|) = \operatorname{tr}(Q^TDQ|V||V|^{-1}) = \operatorname{tr}(Q^TDQ).\end{aligned}$$

Si llamamos $W = Q^T$, lo que hemos obtenido es que la maximización del problema 2.8 equivale a la maximización en $W$ de $\operatorname{tr}(WDW^T)$, sujeto a que $WW^T = I$, puesto que $Q^TQ = I$, obteniendo justamente el problema del teorema 2.4.6. Si suponemos la diagonal de $D$ ordenada de mayor a menor, una matriz $W$ que resuelve este problema se obtiene añadiendo por filas los vectores $e_1, \dots, e_{d'}$ de la base canónica de $\mathbb{R}^d$. Entonces $Q$ contiene los mismos vectores, pero por columnas. Observemos que el valor de la traza del cociente $T(X) = \operatorname{tr}\left((X^TBX)^{-1}(X^TAX)\right)$, con $X \in \mathcal{M}_{d\times d'}(\mathbb{R})$ no varía al multiplicar a la derecha por una matriz regular. En efecto, si $R \in \mathrm{GL}_{d'}(\mathbb{R})$,

$$\begin{aligned}T(XR) = tr\left((R^TX^TBXR)^{-1}(R^TX^TAXR)\right) &= \operatorname{tr}(R^{-1}(X^TBX)^{-1}R^{-T}R^T(X^TAX)R)\\ = \operatorname{tr}((X^TBX)^{-1}(X^TAX)RR^{-1}) &= T(X).\end{aligned}$$

Como $U$ maximiza $T$ y $U = PQ|V|$, se tiene que $PQ$ también maximiza $T$. Además, como de $P^TAP = D$ y $P^TBP = I$ se obtiene que

$$D = P^TAP = (P^TBP)^{-1}(P^TAP) = P^{-1}B^{-1}P^{-T}P^TAP = P^{-1}B^{-1}AP,$$

se concluye que $P$ diagonaliza a $B^{-1}A$ y, por tanto, contiene por columnas los vectores propios de dicha matriz. Como $Q$ contiene los $d'$ primeros vectores propios de la base usual por columnas, $PQ$ tiene por columnas los primeros vectores propios de $B^{-1}A$, que son los asociados a los mayores valores propios. Esto termina la prueba, pues entonces una solución del problema 2.8, que es igual que el de maximizar $T$ salvo una traspocición, consiste en añadir estos vectores por filas. □

## 2.5. Últimas consideraciones

### 2.5.1. Producto tensorial de vectores

**Definición 2.5.1.** Sean $x, y \in \mathbb{R}^d$. Se define el *producto tensorial* de $x$ e $y$ como

$$x \otimes y = xy^T \in \mathcal{M}_d(\mathbb{R}).$$

El producto tensorial de vectores es también conocido en inglés como *outer product*, en contraposición con el *inner product*, que hace referencia al producto escalar usual $\langle x, y\rangle = x^T y$. Observemos que el elemento $(i, j)$ de este producto viene dada por $(x \bigotimes y)_{ij} = x_i y_j$. Una propiedad importante es que esta matriz tiene siempre rango menor o igual que 1, siendo este 1 si y solo $x$ e $y$ son distintos de 0. De hecho, cuando hay alguna fila (resp. columna) no nula, el resto de filas (resp. columnas) son proporcionales a ella.

Un último detalle a destacar es que el producto $xx^T$, para $x \in \mathbb{R}^d \setminus \{0\}$, es siempre semidefinido positivo de rango 1. En efecto, dado $v \in \mathbb{R}^d$, $v^T xx^T v = \langle v, x\rangle\langle x, v\rangle = \langle v, x\rangle^2 \geq 0$. El producto tensorial de vectores será de gran importancia, ya que aparecerá en el cálculo de muchos gradientes, como veremos a continuación, y también formará parte de numerosas matrices de covarianza, compacidad o dispersión que veremos en próximos capítulos.

### 2.5.2. Optimización matricial

Gracias a la norma de Frobenius introducida sobre las matrices, podemos establecer una teoría de derivación en $\mathcal{M}_{d'\times d}(\mathbb{R})$ idéntica a la de $\mathbb{R}^{d'\times d}$. De esta forma, podemos extender todos los resultados de derivación a matrices. Un concepto que será de gran interés será el de gradiente de una función $f\colon \mathcal{M}_{d'\times d}(\mathbb{R}) \to \mathbb{R}$ diferenciable. Lo podemos construir, al igual que en $\mathbb{R}^{d'\times d}$ a partir de las derivadas parciales de $f$, vista la matriz variable como un vector. Sin embargo, debemos conservar la estructura matricial para el gradiente, para ser tratable en el espacio de matrices. Por ello, el gradiente de $f$ en el punto $M = (M_{ij})_{\substack{i\in\{1,\dots,d\}\\ j\in\{1,\dots,d'\}}}$, vendrá dado por la matriz $\nabla f(M) \in \mathcal{M}_{d'\times d}(\mathbb{R})$, donde

$$(\nabla f(M))_{ij} = \frac{\partial f}{\partial M_{ij}}(M),$$

donde las derivadas parciales respecto de $M_{ij}$ se toman en el sentido del análisis real. El gradiente así definido conserva todas las propiedades vectoriales de $\mathbb{R}^{d'\times d}$ (en particular, la regla de adaptación del gradiente descendente sigue siendo válida), y permite además trabajar con él a través del producto de matrices. Será interesante conocer algunas reglas de derivación.

**Proposición 2.5.1.**

a) *Sea* $f\colon \mathcal{M}_{d'\times d}(\mathbb{R})$ *dada por* $f(L) = \|Lx\|_2^2$. *Entonces,* $\nabla f(L) = 2Lxx^T$.

b) *Sea* $f\colon S_d(\mathbb{R}) \to \mathbb{R}$ *dada por* $f(M) = x^T My$. *Entonces,* $\nabla f(M) = xy^T$.

c) *Sea* $f\colon \mathcal{M}_d(\mathbb{R})^+ \to \mathbb{R}$ *dada por* $f(M) = \log\det(M)$. *Entonces,* $\nabla f(M) = M^{-T} = M^{-1}$.

*Demostración.* Los dos primeros apartados se comprueban fácilmente desarrollando $f$ en función de los elementos de la matriz, y tomando derivadas parciales. Para el último caso hay que considerar el desarrollo del determinante por adjuntos. De aquí es fácil observar que el gradiente del determinante de una matriz $M$ es la matriz adjunta, $M^*$. Al componer con el logaritmo, se obtiene que $\nabla f(M) = M^*/\det(M) = M^{-T}$. La última igualdad se tiene por ser $M$ simétrica. □

# Capítulo 3

# Teoría de la información y divergencias

En este capítulo se hará una breve introducción a la teoría de la información. La teoría de la información proporciona una base teórica para muchos campos de la ciencia e ingeniería, especialmente en la informática, las telecomunicaciones o las probabilidades. Dentro de la teoría de la información nos centraremos en el estudio de las divergencias. El concepto de divergencia, asociado a distribuciones de probabilidad, es similar al de distancia, estableciendo una medida para determinar la cercanía entre distribuciones. Dicha medida será de gran utilidad en los algoritmos que estudiaremos en el capítulo 6. También veremos cómo estas divergencias permiten tratar la distribución normal multivariante matricialmente, pudiendo así hacer uso de las teorías desarrolladas en el capítulo 2.

## 3.1. Introducción

La teoría de la información es una rama de las matemáticas y de la teoría de la computación, cuya finalidad es establecer una medida rigurosa para cuantificar la información y el desorden presente en un mensaje de comunicación. Fue desarrollada con el objetivo de encontrar límites en las operaciones de procesamiento de señales como la compresión, el almacenamiento o la comunicación. En la actualidad, sus aplicaciones se extienden a la mayoría de campos de la ciencia y la ingeniería.

El concepto de información puede resultar bastante abstracto, y difícil de modelar con una única definición. La teoría de la información define la entropía como primer concepto para modelar la información. La entropia se define para una distribución de probabilidad, y mide la cantidad de incertidumbre de la fuente de información asociada a dicha distribución. Puede entenderse también como la cantidad de información que contiene una variable aleatoria sobre sí misma. Formalmente, su definición es la que se muestra a continuación.

**Definición 3.1.1.** Sea $(\Omega, \mathcal{A}, P)$ un espacio de probabilidad, y $X\colon \Omega \to \mathbb{R}$ una variable aleatoria, discreta o continua, en dicho espacio. Supongamos que $p$ es la función masa de probabilidad asociada, si $X$ es discreta, o la función de densidad, si $X$ es continua. Se define la *entropía* asociada a la variable aleatoria $X$ como

$$H(x) = \mathbb{E}[1/\log(p(X))] = \mathbb{E}[-\log p(X)] = -\int_{\Omega} \log(p(X(\omega)))\, dP,$$

siempre que dicha integral exista. En particular, si $X$ es discreta y su función masa de probabilidad viene dada por $p(x) = P[X = x]$, entonces se tiene que

$$H(x) = -\sum_{x \in X(\Omega)} p(x)\log(p(x))$$

Si $X$ es una variable aleatoria continua, entonces la entropía de $X$ puede expresarse, siempre que el valor absoluto asociado tenga integral finita, como

$$H(x) = -\int_{-\infty}^{+\infty} p(x)\log(p(x))\ dx$$

Por razones de continuidad, se asume que $0\log 0 = 0$.

Es inmediato comprobar que $H$ toma valores no negativos. También, si cambiamos la base del logaritmo, la entropía difiere únicamente en una constante, por lo que en muchos casos, según el campo de aplicación, se trabaja con una determinada base en la definición de entropía. Si se elige base 2, la entropía se mide en *bits*. A lo largo del capítulo se trabajará en base en $e$. En este caso, la entropía se mide en *nats*.

La entropía mide la cantidad de incertidumbre o información esperada en un suceso. Por ejemplo, si una variable aleatoria toma dos valores con probabilidades $p$ y $1-p$, la entropía será máxima para $p = 1/2$, y será nula cuando $p \in \{0, 1\}$. También, en base 2, es posible ver la entropía, sobre una variable discreta, como una aproximación del número esperado de preguntas binarias de la forma "¿$X = x_i$?" necesarias para acertar el valor que ha tomado $X$. Por tanto, si $X$ tiene una distribución parecida a la uniforme, su entropía será mayor frente a otras distribuciones más deterministas.

Asociados a la entropía se definen numerosos conceptos, como la información mutua, que mide la cantidad de información que contiene una variable aleatoria sobre otra variable, o la entropía relativa, que es una forma de medir la cercanía entre distintas variables aleatorias. Nos centraremos en esta última, y en conceptos derivados de ella. Para ello, en primer lugar, definiremos el concepto de divergencia. La divergencia es una magnitud para medir la cercanía entre determinados objetos sobre un conjunto. No hablaremos de distancias, pues las magnitudes que consideraremos pueden no verificar algunas de las propiedades que se exigen a las distancias, como es el caso de la simetría, o la desigualdad triangular.

**Definición 3.1.2.** Sea $X$ un conjunto. Una aplicación $D(\cdot\|\cdot)\colon X\times X \to \mathbb{R}$ se dice que es una *divergencia* si satisface las siguientes propiedades:

a) No negatividad: $D(x\|y) \geq 0$ para todos $x, y \in X$.

b) Coincidencia: $D(x\|y) = 0$ si y solo si $x = y$.

## 3.2. Las divergencias de Kullback-Leibler y Jeffrey

En esta sección presentaremos las divergencias asociadas a la entropía con las que vamos a trabajar.

**Definición 3.2.1.** Sea $(\Omega, \mathcal{A}, P)$ un espacio de probabilidad, y $X\colon \Omega \to \mathbb{R}$ una variable aleatoria, discreta o continua, en dicho espacio. Supongamos que $p$ es la función masa de probabilidad asociada, si $X$ es discreta, o la función de densidad, si $X$ es continua. Supongamos que $q$ es otra función masa de probabilidad o función de densidad. Entonces, se define la *entropía relativa* o *divergencia de Kullback-Leibler* entre $p$ y $q$, como

$$\mathrm{KL}(p\|q) = \mathbb{E}_p\left[\log\frac{p(X)}{q(X)}\right] = \int_\Omega \log\frac{p(X(\omega))}{q(X(\omega))}\ dP,$$

siempre que dicha integral exista. En particular, si $p$ y $q$ son discretas y toman valores en los mismos puntos, se tiene que

$$\mathrm{KL}(p\|q) = \sum_{x\in X(\Omega)} p(x)\log\frac{p(x)}{q(x)},$$

y si $p$ y $q$ son continuas, siempre que el valor absoluto tenga integral finita, se tiene que

$$\mathrm{KL}(p\|q) = \int_{-\infty}^{+\infty} p(x)\log\frac{p(x)}{q(x)}\,dx.$$

De nuevo por razones de continuidad, se asume que $0\log(0/0) = 0$.

Mediante la entropía relativa se mide la ineficiencia de asumir que $X$ sigue una distribución $q$ cuando la distribución real es $p$. Veamos que, efectivamente, esta aplicación es una divergencia (en el caso continuo, entenderemos la igualdad como igualdad c.p.d.).

**Teorema 3.2.1** (Desigualdad de la información)**.** *La divergencia de Kullback-Leibler es una divergencia, es decir,* $\mathrm{KL}(p\|q) \geq 0$ *y se da la igualdad si y solo si* $p(x) = q(x)$ *c.p.d. en* $X(\Omega)$ *(la igualdad es en todo punto en el caso discreto).*

*Demostración.* Como la función $-\log$ es estrictamente convexa, podemos aplicar la desigualdad de Jensen 1.2.5, obteniendo

$$\begin{aligned}\mathrm{KL}(p\|q) &= \mathbb{E}_p\left[\log\frac{p(X)}{q(X)}\right] = \mathbb{E}_p\left[-\log\frac{q(X)}{p(X)}\right]\\ &\geq -\log\mathbb{E}_p\left[\frac{q(X)}{p(X)}\right] = -\log\int p(x)\frac{q(x)}{p(x)}\,dx\\ &= -\log\int q(x)\,dx = -\log 1 = 0.\end{aligned}$$

Aunque se ha usado la notación continua, el caso discreto es análogo. Además, la convexidad estricta implica que se da la igualdad si y solo si $p/q$ es constante c.p.d. si y solo si $p = q$ c.p.d., por ser funciones de densidad o masa de probabilidad. Además, como en el caso discreto $p$ y $q$ toman valores en conjuntos con probabilidad no nula, se tiene la igualdad en todo punto. □

Tenemos por tanto que, mediante la divergencia de Kullback-Leibler podemos medir la "cercanía" entre dos distribuciones de probabilidad, por lo que podrán ser utilizadas más adelante para acercar distribuciones convenientemente parametrizadas. En cambio, no es tan útil para medir la "lejanía" entre distribuciones, puesto que al no ser simétrica, los valores de $\mathrm{KL}(p\|q)$ y $\mathrm{KL}(q\|p)$ pueden ser muy diferentes cuando $p$ y $q$ no se parecen. Por eso, a veces es útil trabajar con una divergencia de Kullback-Leibler simetrizada. Esta divergencia se conoce como divergencia de Jeffrey.

**Definición 3.2.2.** Se define la divergencia de Jeffrey para dos distribuciones $p$ y $q$ para las que existan $\mathrm{KL}(p\|q)$ y $\mathrm{KL}(q\|p)$ como

$$\mathrm{JF}(p\|q) = \mathrm{KL}(p\|q) + \mathrm{KL}(q\|p).$$

En el caso discreto, se tiene que

$$\mathrm{JF}(p\|q) = \sum_{x=1}^{N}(p(x_i) - q(x_i))(\log p(x_i) - \log q(x_i)).$$

Para el caso continuo, se verifica que

$$\mathrm{JF}(p\|q) = \int_{-\infty}^{+\infty}(p(x) - q(x))(\log p(x) - \log q(x))\,dx.$$

Por la desigualdad de la información es inmediato ver que la divergencia de Jeffrey es efectivamente una divergencia, y además, $J(p\|q) = J(q\|p)$. Observemos que ambas divergencias son funciones de las distribuciones de probabilidad, dependiendo únicamente de las probabilidades establecidas en ellas, y no de los valores que tomen las correspondientes variables aleatorias. Esto permite extender las divergencias a vectores aleatorios, siempre que conozcamos sus funciones de densidad o masa de probabilidad.

## 3.3. La distribución normal multivariante y divergencias matriciales.

Para concluir la sección, vamos a estudiar las aplicaciones de las divergencias de Kullback-Leibler y de Jeffrey sobre la familia de distribuciones gaussianas multivariante. Estas distribuciones son la generalización natural de la distribución normal de una variable. Sea $\mu \in \mathbb{R}^d$ y $\Sigma \in \mathcal{M}_d(\mathbb{R})^+$ una matriz definida positiva. Se dice que un vector aleatorio $X = (X_1, \dots, X_d)$ sigue una distribución normal multivariante con media $\mu$ y matriz de covarianza $\Sigma$, si tiene una función de densidad asociada

$$p(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2} \det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right).$$

Se verifica que $\mathbb{E}[X] = \mu$ y $\mathrm{Cov}(X) = \mathbb{E}[(X - \mathbb{E}[X])(X - \mathbb{E}[X])^T] = \Sigma$, y por tanto las distribuciones gaussianas quedan determinadas por su media y su covarianza. Es importante recordar que, si $L \in \mathcal{M}_d(\mathbb{R})$ es una aplicación lineal, entonces, el vector aleatorio $LX$ sigue una distribución normal multivariante de media $L\mu$ y covarianza $L\Sigma L^T$.

Queremos establecer, para distribuciones normales multivariante, correspondencias entre las divergencias estudiadas y divergencias matriciales. Las divergencias matriciales proporcionan una alternativa para medir la cercanía entre matrices a la ya estudiada en el capítulo 2 con la norma de Frobenius. Nos centraremos en las conocidad como divergencias de Bregman.

**Definición 3.3.1.** Sea $K \subset \mathcal{M}_d(\mathbb{R})$ un conjunto convexo y abierto, y $\phi\colon K \to \mathbb{R}$ una función estrictamente convexa y diferenciable. Se define la *divergencia de Bregman* asociada a $\phi$ a la aplicación $D_\phi(\cdot\|\cdot)\colon K \times K \to \mathbb{R}$ dada por

$$D_\phi(A\|B) = \phi(A) - \phi(B) - \mathrm{tr}(\nabla\phi(B)^T(A-B)) = \phi(A) - \phi(B) - \langle \nabla\phi(B), B - A\rangle_F.$$

Es interesante destacar que para $\phi(X) = \|X\|_F^2$, la divergencia de Bregman asociada es $D_\phi(A\|B) = \|A-B\|_F^2$, es decir, la distancia asociada a la norma de Frobenius estudiada en el capítulo anterior. Es claro que la divergencia de Bregman es una divergencia, para cualquier $\phi$ en las condiciones de la definición, gracias al apartado d) de la proposición 1.2.2.

De todas las posibles divergencias de Bregman, nos centraremos en aquella asociada a la función *log-det*, es decir, la función $\phi_{ld}\colon \mathcal{M}_d(\mathbb{R})^+ \to \mathbb{R}$ dada por

$$\phi_{ld}(M) = -\log\det(M) = -\log\prod_{i=1}^d \lambda_i = -\sum_{i=1}^d \log\lambda_i,$$

donde $0 < \lambda_1 \leq \cdots \leq \lambda_d$ son los valores propios de $M$.

**Proposición 3.3.1.** *$\phi_{ld}$ es estrictamente convexa en $\mathcal{M}_d(\mathbb{R})^+$.*

*Demostración.* Fijadas $M_0, M \in \mathcal{M}_d(\mathbb{R})^+$, definimos $g\colon [0,1] \to \mathbb{R}$ por $g(t) = -\phi_{ld}(M_0 + tM) = \log\det(M_0 + tM)$. Extrayendo raíces cuadradas a izquierda y derecha, y usando las propiedades del determinante y logaritmo, obtenemos

$$\begin{aligned}\log\det(M_0 + tM) &= \log\det(M_0^{1/2}(I + tM_0^{1/2}MM_0^{1/2})M_0^{1/2}) \\ &= \log(\det(I + tM_0^{1/2}MM_0^{1/2})\det(M_0^{1/2})\det(M_0^{1/2}) \\ &= \sum_{i=1}^{d}\log(1+t\sigma_i) + \log\det(M_0),\end{aligned}$$

donde $\sigma_1 \leq \cdots \leq \sigma_d$ son los valores propios de $N = M_0^{1/2}MM_0^{1/2}$, los cuales son todos positivos pues es una $N$ y $M$ son congruentes, $M$ es definida positiva, y la congruencia preserva el signo de los valores propios. Entonces,

$$g'(t) = \sum_{i=1}^{d}\frac{\sigma_i}{1+t\sigma_i} \quad g''(t) = -\sum_{i=1}^{d}\frac{\sigma_i^2}{(1+t\sigma_i)^2} < 0 \quad \forall t \in [0,1].$$

Por tanto, $g$ es estrictamente cóncava, luego $-\phi_{ld}$ es estrictamente cóncava y $\phi_{ld}$ es estrictamente convexa. □

Como consecuencia de la proposición, $\phi_{ld}$ determina una divergencia de Bregman, la cual se denomina divergencia *log-det* y se nota $D_{ld}$. Podemos calcular su expresión, para $A, B \in \mathcal{M}_d(\mathbb{R})^+$, obteniendo

$$D_{ld}(A\|B) = \log\det(B) - \log\det(A) + \langle B^{-1}, A - B\rangle_F = \operatorname{tr}(AB^{-1}) - \log\det(AB^{-1}) - d.$$

Si llamamos $a_1, \ldots, a_d$ a los vectores propios de $A$, con correspondientes valores propios $\lambda_1, \ldots, \lambda_d$ y llamamos $b_1, \ldots, b_d$ a los vectores propios de $B$ con valores propios $\mu_1, \ldots, \mu_d$, se tiene entonces la expresión

$$D_{ld}(A\|B) = \sum_{i=1}^{d}\sum_{j=1}^{d}\frac{\lambda_i}{\mu_j}\langle a_i, b_j\rangle^2 - \sum_{i=1}^{d}\log\frac{\lambda_i}{\mu_i} - d.$$

Vamos finalmente a calcular la entropía entre dos distribuciones normales. Sean $\mu_1, \mu_2 \in \mathbb{R}^d$ y $\Sigma_1, \Sigma_2 \in \mathcal{M}_d(\mathbb{R})^+$. Consideramos las distribuciones gaussianas asociadas a $(\mu_1, \Sigma_1)$ y $(\mu_2, \Sigma_2)$, respectivamente, determinadas por sus funciones de densidad $p(x|\mu_1, \Sigma_1)$ y $p(x|\mu_2, \Sigma_2)$. Buscamos calcular $\mathrm{KL}(p(x|\mu_1, \Sigma_1)\|p(x|\mu_2, \Sigma_2))$. Si abreviamos las distribuciones por $p_1$ y $p_2$, respectivamente, tenemos, en primer lugar, que

$$\mathrm{KL}(p_1\|p_2) = \int p_1(x)\log\frac{p_1(x)}{p_2(x)}\,dx = \int p_1(x)\log p_1(x) - \int p_1(x)\log p_2(x) = -H(p_1) - \int p_1\log p_2(x).$$

Podemos calcular la entropía asociada a $p_1$, como

$$\begin{aligned}
H(p_1) &= -\int p_1(x)\left[\frac{-1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1) - \log((2\pi)^{d/2}\det(\Sigma_1)^{1/2})\right]\,dx \\
&= \frac{1}{2}\mathbb{E}_{p_1}\left[\sum_{i,j=1}^{d}(x_i-\mu_i^{(1)})(\Sigma_1^{-1})_{ij}(x_j-\mu_j^{(1)})\right] + \frac{1}{2}\log((2\pi)^d\det(\Sigma_1)) \\
&= \frac{1}{2}\sum_{i,j=1}^{d}\mathbb{E}_{p_1}[(x_j-\mu_j^{(1)})(x_i-\mu_i^{(1)})](\Sigma_1^{-1})_{ij} + \frac{1}{2}\log((2\pi)^d\det(\Sigma_1)) \\
&= \frac{1}{2}\sum_{j=1}^{d}\sum_{i=1}^{d}(\Sigma_1)_{ji}(\Sigma_1^{-1})_{ij} + \frac{1}{2}\log((2\pi)^d\det(\Sigma_1)) \\
&= \frac{1}{2}\sum_{j=1}^{d}(\Sigma_1\Sigma_1^{-1})_{jj} + \frac{1}{2}\log((2\pi)^d\det(\Sigma_1)) \\
&= \frac{1}{2}\sum_{j=1}^{d}1 + \frac{1}{2}\log((2\pi)^d\det(\Sigma_1)) \\
&= \frac{d}{2} + \frac{1}{2}\log((2\pi)^d\det(\Sigma_1)),
\end{aligned} \tag{3.1}$$

donde para el cuarto paso se ha utilizado la definición de matriz de covarianza, y para el quinto paso se ha utilizado la definición elemento a elemento del producto de matrices. Calculamos ahora el segundo sumando de la expresión de la divergencia.

$$\begin{aligned}
\int p_1(x)\log p_2(x) &= \int p_1(x)\left[-\frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2) - \log((2\pi)^{d/2}\det(\Sigma_2)^{1/2})\right]\,dx \\
&= -\frac{1}{2}\sum_{i,j=1}^{d}(\Sigma_2^{-1})_{ij}\mathbb{E}_{p_1}[(x_j-\mu_j^{(2)})(x_i-\mu_i^{(2)})] - \frac{1}{2}\log((2\pi)^d\det(\Sigma_2)) \\
&= -\frac{1}{2}\operatorname{tr}(\Sigma_2^{-1}\mathbb{E}_{p_1}[(x-\mu_2)(x-\mu_2)^T]) - \frac{1}{2}\log((2\pi)^d\det(\Sigma_2)) \\
&= -\frac{1}{2}\operatorname{tr}(\Sigma_2^{-1}\mathbb{E}_{p_1}[((x-\mu_1)-(\mu_1-\mu_2))((x-\mu_1)-(\mu_1-\mu_2))^T]) - \frac{1}{2}\log((2\pi)^d\det(\Sigma_2)) \\
&= -\frac{1}{2}\operatorname{tr}(\Sigma_2^{-1}\Sigma_1 + \Sigma_2^{-1}(\mu_1-\mu_2)(\mu_1-\mu_2)^T) - \frac{1}{2}\log((2\pi)^d\det(\Sigma_2)) \\
&= -\frac{1}{2}\operatorname{tr}(\Sigma_2^{-1}\Sigma_1) - \frac{1}{2}(\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2) - \frac{1}{2}\log((2\pi)^d\det(\Sigma_2)),
\end{aligned} \tag{3.2}$$

donde en el quinto paso se ha utilizado, de nuevo, la definición de matriz de covarianza, y que $\mathbb{E}_{p_1}[(x-\mu_1)(\mu_1-\mu_2)^T] = \mathbb{E}_{p_1}[x-\mu_1](\mu_1-\mu_2)^T = 0$. Combinando las expresiones obtenidas en 3.1 y 3.2, obtenemos

$$\begin{aligned}
\mathrm{KL}(p_1\|p_2) &= -\frac{d}{2} - \frac{1}{2}\log((2\pi)^d\det(\Sigma_1)) + \frac{1}{2}\operatorname{tr}(\Sigma_2^{-1}\Sigma_1) + \frac{1}{2}(\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2) + \frac{1}{2}\log((2\pi)^d\det(\Sigma_2)) \\
&= \frac{1}{2}\left(\operatorname{tr}(\Sigma_1\Sigma_2^{-1}) - \log\det(\Sigma_1\Sigma_2^{-1}) - d\right) + \frac{1}{2}(\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2) \\
&= \frac{1}{2}D_{ld}(\Sigma_1\|\Sigma_2) + \frac{1}{2}\|\mu_1-\mu_2\|^2_{\Sigma_2^{-1}}.
\end{aligned} \tag{3.3}$$

Acabamos de obtener el siguiente resultado.

**Teorema 3.3.2.** *La divergencia de Kullback-Leibler entre dos distribuciones gaussianas multivariante definidas por funciones de densidad $p_1(x|\mu_1,\Sigma_1)$ y $p_2(x|\mu_2,\Sigma_2)$, con $\mu_1,\mu_2\in\mathbb{R}^d$ y $\Sigma_1,\Sigma_2\in\mathcal{M}_d(\mathbb{R})^+$, verifica*

$$\mathrm{KL}(p_1\|p_2)=\frac{1}{2}D_{ld}(\Sigma_1\|\Sigma_2)+\frac{1}{2}\|\mu_1-\mu_2\|^2_{\Sigma_2^{-1}}.$$

**Corolario 3.3.3.** *La divergencia de Kullback-Leibler entre dos distribuciones gaussianas multivariante definidas por funciones de densidad $p_1$ y $p_2$ con igual media, y covarianzas $\Sigma_1$ y $\Sigma_2$, verifica*

$$\mathrm{KL}(p_1\|p_2)=\frac{1}{2}D_{ld}(\Sigma_1\|\Sigma_2).$$

De forma inmediata se obtienen expresiones para la divergencia de Jeffrey.

**Corolario 3.3.4.** *La divergencia de Jeffrey entre dos distribuciones gaussianas multivariante definidas por funciones de densidad $p_1(x|\mu_1,\Sigma_1)$ y $p_2(x|\mu_2,\Sigma_2)$, con $\mu_1,\mu_2\in\mathbb{R}^d$ y $\Sigma_1,\Sigma_2\in\mathcal{M}_d(\mathbb{R})^+$, verifica*

$$\mathrm{JF}(p_1\|p_2)=\frac{1}{2}\operatorname{tr}(\Sigma_1^{-1}\Sigma_2+\Sigma_1^{-1}\Sigma_2)-d+\frac{1}{2}\|\mu_1-\mu_2\|^2_{\Sigma_1^{-1}+\Sigma_2^{-1}}.$$

*Demostración.*

$$\begin{aligned}
\mathrm{JF}(p_1\|p_2)&=\mathrm{KL}(p_1\|p_2)+\mathrm{KL}(p_2\|p_1)\\
&=\frac{1}{2}D_{ld}(\Sigma_1\|\Sigma_2)+\frac{1}{2}(\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2)+\frac{1}{2}D_{ld}(\Sigma_2\|\Sigma_1)+(\mu_1-\mu_2)^T\Sigma_1^{-1}(\mu_1-\mu_2)\\
&=\frac{1}{2}\left[\operatorname{tr}(\Sigma_1\Sigma_2^{-1})+\operatorname{tr}(\Sigma_1^{-1}\Sigma_2)\right]-\frac{1}{2}\left[\log\det(\Sigma_1\Sigma_2^{-1})+\log\det(\Sigma_1^{-1}\Sigma_2)\right]\\
&+\frac{1}{2}(\mu_1-\mu_2)^T(\Sigma_1^{-1}+\Sigma_2^{-1})(\mu_1-\mu_2)-\frac{1}{2}(d+d)\\
&=\frac{1}{2}\operatorname{tr}(\Sigma_1\Sigma_2^{-1}+\Sigma_1^{-1}\Sigma_2)-d+\frac{1}{2}\|\mu_1-\mu_2\|^2_{\Sigma_1^{-1}+\Sigma_2^{-1}}-\frac{1}{2}(\log\det\Sigma_1-\log\det\Sigma_2+\log\det\Sigma_2-\log\det\Sigma_1)\\
&=\frac{1}{2}\operatorname{tr}(\Sigma_1^{-1}\Sigma_2+\Sigma_1^{-1}\Sigma_2)-d+\frac{1}{2}\|\mu_1-\mu_2\|^2_{\Sigma_1^{-1}+\Sigma_2^{-1}}.
\end{aligned}$$

□

**Corolario 3.3.5.** *La divergencia de Jeffrey entre dos distribuciones gaussianas multivariante definidas por funciones de densidad $p_1$ y $p_2$ con igual media, y covarianzas $\Sigma_1$ y $\Sigma_2$, verifica*

$$\mathrm{JF}(p_1\|p_2)=\frac{1}{2}\operatorname{tr}(\Sigma_1^{-1}\Sigma_2+\Sigma_1^{-1}\Sigma_2)-d.$$

# Parte II

# Informática teórica

# Capítulo 4

# El aprendizaje automático

En este capítulo se realiza una descripción general del aprendizaje automático, haciendo especial hincapié en el aprendizaje supervisado y los problemas de clasificación. El aprendizaje automático es la rama de las ciencias de la computación cuya finalidad es conseguir que los ordenadores aprendan de un conjunto de datos. Aprender en este contexto está relacionado con la identificación de patrones o la capacidad de generalización a nuevos datos. Veremos que hay diferentes modelos de aprendizaje, distinguiendo dos grandes grupos: supervisado y no supervisado. Profundizando en el aprendizaje supervisado, analizaremos los problemas de clasificación, sobre los que se centrarán la mayoría de las aplicaciones del problema que estudiaremos en los próximos capítulos.

## 4.1. Introducción

El objetivo del aprendizaje automático es, como se ha indicado al comienzo del capítulo, aprender de los datos. Este aprendizaje se realiza mediante la elaboración de algoritmos de aprendizaje. Tradicionalmente, un algoritmo clásico recibe como entrada un conjunto de datos y produce como salida valores en función a dichos datos de entrada. En los algoritmos de aprendizaje, la técnica difiere notablemente. En este caso, el algoritmo recibe como entrada un conjunto de datos (y, posiblemente, un conjunto de valores asociados a esos datos), que se denominan datos de entrenamiento, y produce como salida una función o un algoritmo aplicable a nuevos datos. Esta función es el resultado del aprendizaje realizado con los datos de entrada.

El aprendizaje automático surge cuando la complejidad o la variabilidad del problema a resolver impiden la elaboración de un programa que pueda resolverlo de forma determinista o heurística. Entre este tipo de problemas nos encontramos:

- *Realizar tareas humanas o animales.* Hay muchas tareas que los seres humanos realizamos diariamente, pero sin tener la suficiente consciencia de cómo las hacemos, de forma que no disponemos del suficiente conocimiento como para extraer un algoritmo que pueda ejecutarlas automáticamente. Entre este tipo de tareas se encuentran, por ejemplo, la conducción y el reconocimiento de sonidos o imágenes. En estos casos, son de gran utilidad los programas que sean capaces de aprender de experiencias previas.
- *Tareas inabordables por el ser humano.* Este tipo de tareas está relacionado con el análisis de conjuntos de datos de gran tamaño y la búsqueda de patrones en ellos. En la era digital que vivimos actualmente, es posible disponer de inmensas cantidades de datos en cualquier ámbito. Aprender de estos datos para convertirlos en conocimiento es de gran importancia en campos como la medicina, la meteorología, la biología y la administración de empresas.

- *Variablilidad de los problemas.* Los algoritmos clásicos son rígidos, en el sentido de que una vez creados, permanecen invariables a lo largo del tiempo. Sin embargo, la realización de determinadas tareas puede variar con el tiempo, con los factores del entorno, o según quién la realiza. Problemas de estas características son el reconocimiento de dígitos escritos a mano o la detección de spam en los correos electrónicos. Los algoritmos de aprendizaje son, por su naturaleza, capaces de adaptarse a esta variabilidad.

Aunque existen distintas formas de clasificar los problemas de aprendizaje automático, la clasificación más popular divide estos problemas principalmente en dos grupos: aprendizaje no supervisado y aprendizaje supervisado.

En el aprendizaje supervisado el algoritmo de aprendizaje dispone de un conjunto de datos de entrenamiento para los cuales se conoce el valor de la función que se quiere aprender. Se dice en este caso que los datos están etiquetados. A partir de los datos etiquetados, se busca una función capaz de etiquetar nuevos datos. Estudiaremos este problema en la siguiente sección.

En el aprendizaje no supervisado los datos no están etiquetados, es decir, solo disponemos de la información presente en los datos, sin saber cómo debe actuar la función que queremos aprender sobre estos. Dentro de estos problemas, destaca el de segmentación o *clustering*, que trataremos brevemente en alguno de los algoritmos que estudiaremos en los próximos capítulos. El problema de clustering consiste en, dado el conjunto de datos de entrenamiento sin etiquetar, encontrar una forma de agrupar los datos en distintos *clusters*, de forma que los datos en un mismo cluster sean similares, y a su vez diferentes a los datos en distintos clusters. Otro problema interesante de aprendizaje no supervisado es el aprendizaje de reglas de asociación, que consiste en extraer relaciones de causa y consecuencia en el conjunto de datos, obteniendo así reglas que permitan interpretarlos.

## 4.2. El aprendizaje supervisado

Como ya se ha comentado, el aprendizaje supervisado consiste en aprender de datos etiquetados. Existen diversas formas de modelar este problema [7, 31]. Siguiendo el enfoque de [31], los elementos de un algoritmo de aprendizaje supervisado son:

- **La entrada del algoritmo de aprendizaje.** Esta está compuesta por los siguiente elementos:
  - **El dominio del problema.** Es un conjunto arbitrario, denominado usualmente $\mathcal{X}$. Normalmente es producto cartesiano de otros conjuntos, $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$, de forma que sus elementos son vectores, denominados *vectores de atributos*. Un caso importante sucede al tratar con datos numéricos. En esta situación se suele tomar $\mathcal{X} = \mathbb{R}^d$.
  - **El conjunto de etiquetas.** Es otro conjunto arbitrario, denominado $\mathcal{Y}$. Las elecciones más comunes para $\mathcal{Y}$ son conjuntos finitos, destacando el de dos elementos, o, por otro lado, el conjunto de los números reales.
  - **El conjunto de los datos de entrenamiento.** Es una muestra finita $S = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathcal{X} \times \mathcal{Y}$. Son los datos a los que tiene acceso el algoritmo de aprendizaje.
- **La salida del algoritmo de aprendizaje.** El algoritmo de aprendizaje elabora una *regla de predicción*, *predictor* o *hipótesis*, $h\colon \mathcal{X} \to \mathcal{Y}$. Esta función permite predecir la etiqueta de nuevos puntos en el dominio.

- **Un modelo de generación de datos.** Se asume que las instancias de datos se generan siguiendo una determinada distribución de probabilidad sobre el conjunto $\mathcal{X}$. Notamos dicha distribución por $\mathcal{D}$. Los datos de entrenamiento son $N$ muestras de esta distribución. Para el etiquetado, se asume que existe una función "correcta" (y desconocida) de etiquetado, $f\colon \mathcal{X} \to \mathcal{Y}$, de forma que las etiquetas de los datos de entrenamiento son generadas como $y_i = f(x_i)$, para $i = 1, \ldots, N$. Esta función es la que busca encontrar el algoritmo de aprendizaje.
- **Medidas del error.** Asociado al predictor aprendido $h$, se define su error como la probabilidad de que no asigne la etiqueta correcta a un punto aleatorio $x \sim \mathcal{D}$, es decir,

$$L_{\mathcal{D},f}(h) = \mathbb{P}_{x\sim\mathcal{D}}[h(x) \neq f(x)].$$

  Este error se denomina también *error de generalización* o *riesgo* de $h$. Notemos que este error no podemos conocerlo, pues depende de $\mathcal{D}$ y $f$, que son desconocidas. Se harán estimaciones de este error con los datos de entrenamiento.

Aunque los elementos anteriores son los que modelan el aprendizaje supervisado, como se ha comentado en las líneas previas, muchos de los elementos no son conocidos. En particular, no podemos conocer el error real de los predictores que aprendamos. Por ello, es común utilizar otros conceptos de error que dependen únicamente de los datos de entrenamiento, y que por tanto pueden ser calculados por el algoritmo de aprendizaje. Este error dependerá del tipo de problema de aprendizaje supervisado que estemos tratando, y se denomina *error empírico* o *riesgo empírico*, notándose como $L_S$. Aprender predictores que minimicen este error conforma un paradigma de aprendizaje denominado *minimización del riesgo empírico* o ERM.

Sin embargo, la minimización del riesgo empírico puede conducir a predictores erróneos, pues un predictor que se ajuste demasiado a los datos puede perder capacidad de generalización, aprendiendo características específicas de los datos de entrenamiento que no son relevantes en su distribución real. Esto se manifiesta de forma más severa si en los datos hay presente ruido, debido por ejemplo a errores de medición durante la recogida de estos. Este fenómeno se conoce como *sobreaprendizaje*. Una solución usual para evitar este problema consiste en restringir la búsqueda de predictores a un conjunto de hipótesis $\mathcal{H}$ , y seleccionar el predictor dentro de dicha clase que minimice el riesgo empírico. Si se tiene algún conocimiento a priori sobre los datos, puede utilizarse para determinar la clase de hipótesis $\mathcal{H}$ sobre la que aprender.

La minimización del riesgo empírico no es el único paradigma de aprendizaje popular. Existen otros paradigmas, como la minimización del riesgo estructural (SRM), en el cual la selección de hipótesis viene determinada, además de por la minimización del riesgo empírico, por un parámetro de regularización que asigna un peso distinto a determinados subconjuntos de hipótesis, según la simplicidad (y por tanto, capacidad de generalización) de estos. Esta modelización del aprendizaje supervisado se profundiza con mayor detalle dentro de la teoría PAC, en la que se proporciona un marco teórico amplio sobre las capacidades de generalización de los algoritmos de aprendizaje [31].

Para concluir presentamos los problemas más comunes del aprendizaje supervisado.

- **Clasificación.** Es, probablemente, el problema más popular del aprendizaje supervisado. En este caso, el conjunto $\mathcal{Y}$ es finito y, salvo excepciones, no aporta información cardinal ni ordinal. Los distintos elementos que lo componen se denominan clases. Los predictores en este caso se denominan *clasificadores*. El error empírico de los clasificadores se mide como el número de clases no acertadas, normalizado, es decir, $L_S(h) = |\{i \in \{1, \ldots, N\}\colon h(x_i) \neq y_i\}|/N$.

- **Regresión**. En este caso, el conjunto $\mathcal{Y}$ es $\mathbb{R}$. Los datos tienen asociado un valor real como etiqueta, y el predictor aprendido es por tanto una función real que intenta aproximar los datos del conjunto de entrenamiento. En este caso, se utiliza como medida del error empírico el error cuadrático medio, es decir, $L_S(h) = \sum_{i=1}^{N}(h(x_i) - y_i)^2/N$.

## 4.3. El problema de clasificación

El problema de clasificación consiste, por tanto, en aprender una función (clasificador) que asigne a cada dato en el dominio $\mathcal{X}$ una clase de entre un conjunto finito de clases, que conforman $\mathcal{Y}$. El problema de clasificación más común es el binario, donde el conjunto $\mathcal{Y}$ tiene únicamente dos elementos. Se suele representar en estos casos $\mathcal{Y} = \{-1, 1\}$ o $\mathcal{Y} = \{0, 1\}$. La clasificación binaria es de gran importancia, pues cualquier otro problema de clasificación puede reducirse a subproblemas de clasificación binaria. Muchos algoritmos de clasificación trabajan, por tanto, con problemas binarios, y reducen los problemas con más clases a subproblemas de este tipo. Los problemas de clasificación con más de dos clases se conocen como multiclase. Veremos que, para el problema que vamos a tratar, que haya más de dos clases no supondrá ningún inconveniente, no habiendo por tanto necesidad de descomponer el problema.

Los problemas de clasificación tienen gran cantidad de aplicaciones en ámbitos muy variados, como por ejemplo, la detección de enfermedades, el reconocimiento de imágenes o sonidos, la detección de correos de spam, el desarrollo de motores de búsqueda en Internet, la taxonomía dentro de la biología o la clasificación de documentos.

# Capítulo 5

# El aprendizaje de métricas de distancia

En este capítulo se describe el problema central de aprendizaje que se desarrolla en este trabajo. Para ello, se recordarán los conceptos de distancia y pseudodistancia, haciendo especial hincapié en aquellas distancias sobre espacios de Hilbert de dimensión finita. Estas distancias permitirán describir el problema del aprendizaje de métricas de distancia. Finalmente, se describen las aplicaciones de este paradigma de aprendizaje, destacando entre ellas el aprendizaje por semejanza, donde las distancias juegan un papel fundamental.

## 5.1. Distancias. Distancia de Mahalanobis.

### 5.1.1. Definición y ejemplos

El concepto de distancia es fundamental en el paradigma de aprendizaje que vamos a desarrollar. En primer lugar recordamos la definición de distancia y espacio métrico.

**Definición 5.1.1** (Distancia)**.** Sea $X$ un conjuntono vacío. Una *distancia* o *métrica* definida sobre $X$ es una aplicación $d : X \times X \to \mathbb{R}$, verificando las siguientes propiedades:

a) $d(x, y) = 0 \iff x = y$, para todos $x, y \in X$ (Coincidencia)

b) $d(x, y) = d(y, x)$, para todos $x, y \in X$ (Simetría)

c) $d(x, z) \leq d(x, y) + d(y, z)$, para todos $x, y, z \in X$ (Desigualdad triangular)

Al par ordenado $(X, d)$ se le denomina espacio métrico.

**Comentario 5.1.1.** *Como consecuencia de la definición se tienen las siguientes propiedades adicionales.*

d) $d(x, y) \geq 0\ \forall x, y \in X$ *(No negatividad)*

e) $|d(x, y) - d(y, z)| \leq d(x, z)\ \forall x, y, z \in X$ *(Desigualdad triangular por defecto)*

f) $d(x_1, x_n) \leq \sum_{i=1}^{n-1} d(x_i, x_{i+1})\ \forall x_1, \ldots, x_n \in X$ *(Desigualdad triangular generalizada)*

*Demostración.*

d) $0 \underset{a)}{=} \frac{1}{2}d(x, x) \underset{c)}{\leq} \frac{1}{2}[d(x, y) + d(y, x)] \underset{b)}{=} d(x, y)\ \forall x, y \in X$

e) Usando *b)* y *c)* se tiene:

$$d(x, y) \leq d(x, z) + d(z, y) = d(x, z) + d(y, z) \implies d(x, y) - d(y, z) \leq d(x, z)$$

$$d(y,z) \leq d(y,x) + d(x,z) = d(x,y) + d(x,z) \implies d(y,z) - d(x,y) \leq d(x,z)$$

Por tanto podemos tomar valores absolutos en la diferencia obteniendo la desigualdad buscada.

f) Es consecuencia de la desigualdad triangular aplicando inducción.

□

Veamos algunos de los ejemplos más conocidos de espacios métricos.

Ejemplo 5.1.2 (Subespacios métricos): Sea $(X,d)$ un espacio métrico y $A \subset X$. La aplicación $d_{|A} : A \times A \to \mathbb{R}_0^+$ es una distancia, y $(A, d_{|A})$ es un subespacio métrico de $(X,d)$.

Ejemplo 5.1.3 (Distancia trivial): Sea $X$ cualquier conjunto no vacío. Sobre $X$ definimos la aplicación $d : X \times X \to \mathbb{R}_0^+$ por

$$d(x,y) := \begin{cases} 0 & \text{, si } x = y \\ 1 & \text{, si } x \neq y \end{cases}.$$

$(X,d)$ es un espacio métrico, y $d$ es una distancia trivial que nos indica solo si dos elementos de $X$ son iguales o distintos.

Ejemplo 5.1.4 (Distancia de Hamming): Sean $(X_i, d_i), i = 1, \ldots, n$ espacios métricos con distancias triviales. Consideramos $X = X_1 \times \cdots \times X_n$, $x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \in X$ y la aplicación $d : X \times X \to \mathbb{R}_0^+$ dada por

$$d(x,y) = \sum_{i=1}^{n} d(x_i, y_i).$$

La aplicación $d$ es una distancia que nos muestra el número de elementos que difieren entre dos vectores $x, y$ de $X$, y se conoce como distancia de Hamming. Es muy utilizada en algunos ámbitos de la teoría de la información, y es la distancia mas popular sobre espacios cuyas componentes no son ordenables.

Ejemplo 5.1.5 (Distancias asociadas a normas): Si $(X, \|.\|)$ es un espacio normado real, se define la distancia asociada a la norma por $d(x,y) = \|x - y\|$ para todos $x, y \in X$. Las distancias asociadas a normas verifican propiedades adicionales, de combrobación inmediata:

g) $d(ax, ay) = |a| d(x,y)$, para $a \in \mathbb{R}$, $x, y \in X$ (homogeneidad)

h) $d(x,y) = d(x+z, y+z)$ para $x, y, z \in X$ (invarianza por traslaciones)

Profundizaremos sobre estas distancias en las siguientes secciones.

### 5.1.2. Pseudodistancias

El concepto de distancia se puede suavizar, relajando la condición de coincidencia, obteniendo así lo que se conoce como una pseudodistancia, una aplicación que mantiene muchas de las propiedades de una distancia, y en muchos campos, como el que vamos a tratar, puede aplicarse con la misma utilidad que las distancias. De hecho, veremos que en algunos casos una pseudodistancia puede tener propiedades más beneficiosas que una distancia propia. Veamos su definición y algunos ejemplos y propiedades.

**Definición 5.1.2** (Pseudodistancia)**.** Sea $X$ un conjunto no vacío. Una *pseudodistancia* definida sobre $X$ es una aplicación $d : X \times X \to \mathbb{R}_0^+$, verificando las siguientes propiedades:

a) $d(x,x) = 0$, para todo $x \in X$

b) $d(x,y) = d(y,x)$, para todos $x, y \in X$ (Simetría)

c) $d(x,z) = d(x,y) + d(y,z)$, para todos $x, y, z \in X$ (Desigualdad triangular)

Podemos ver que el único cambio de la definición consiste en eliminar una de las implicaciones en la propiedad a) (ahora puede haber elementos distintos con distancia nula entre ellos). Este cambio no afecta a la demostración de las propiedades *d)*, *e)* y *f)* de la distancia, luego estas siguen siendo válidas en las pseudodistancias. Veamos algunos ejemplos de pseudodistancias.

Ejemplo 5.1.6 (Ejemplos básicos): a) Toda distancia sobre $X$ es una pseudodistancia sobre $X$.

b) La aplicación nula $d : X \times X \to \mathbb{R}_0^+$ dada por $d(x,y) = 0 \ \forall x, y \in X$ es una pseudodistancia.

Ejemplo 5.1.7 (Espacios de funciones integrables): Sea $\Omega \subset \mathbb{R}$ y consideramos, para $1 < p < \infty$, los espacios de funciones integrables

$$L^p(\Omega) = \left\{ f \colon \Omega \to \mathbb{R} \colon f \text{ es medible y } \int_\Omega |f(t)|^p dt < \infty \right\}.$$

Dadas $f, g \in L^p(\Omega)$ definimos la pseudodistancia entre ellas como

$$d(f,g) = \left( \int_\Omega |f(t) - g(t)|^p dt \right)^{1/p}.$$

Es claro que se verifican las propiedades *a)* y *b)* de pseudodistancia, y la *c)* es una aplicación directa de la desigualdad integral de Minkowski. Sin embargo, no es una distancia puesto que si $d(f,g) = 0$ solo tenemos asegurada la igualdad casi por doquier.

Ejemplo 5.1.8 (Pseudodistancias asociadas a seminormas): Si $X$ es un espacio vectorial real y $\|.\|$ es una seminorma, se define la distancia asociada a la seminorma por $d(x,y) = \|x - y\| \ \forall x, y \in X$. Estas pseudodistancias verifican también las propiedades *g)* y *h)* que verifican las distancias asociadas a normas. Profundizaremos sobre ellas en la siguiente sección.

Para concluir esta sección, vamos a mostrar que a partir de una pseudodistancia podemos definir una relación de equivalencia mediante la cual, tras identificar los elementos en las mismas clases, podemos obtener un espacio métrico.

**Proposición 5.1.9.** *Sea $X$ un conjunto no vacío y $d : X \times X \to \mathbb{R}_0^+$ una pseudodistancia sobre $X$. Definimos la relación $x \sim y \iff d(x,y) = 0$. $\sim$ es una relación de equivalencia.*
*Demostración.*

- *Reflexiva.* Consecuencia de la propiedad *a)* de pseudodistancia.
- *Simétrica.* Consecuencia de la propiedad *b)* de pseudodistancia.
- *Transitiva.* Consecuencia de la propiedad *c)* de pseudodistancia.

□

**Teorema 5.1.10.** *En las condiciones anteriores, el cociente $X/_\sim$ es un espacio métrico con la distancia $\hat{d} : X/_\sim \times X/_\sim \to \mathbb{R}_0^+$ dada por $\hat{d}([x],[y]) = d(x,y) \ \forall [x],[y] \in X/_\sim$*
*Demostración.* En primer lugar veamos que la aplicación $\hat{d}$ está bien definida. Para ello veamos que la distancia no depende del representante escogido. Supongamos $[x] = [x']$ y $[y] = [y']$ (lo que implica que $d(x,x') = 0 = d(y,y')$. Queremos ver que $d(x,y) = d(x',y')$. Aplicamos varias veces la desigualdad triangular.

$$d(x,y) \leq \underbrace{d(x,x')}_{=0} + d(x',y) \leq d(x',y') + \underbrace{d(y',y)}_{=0}$$
$$\leq \underbrace{d(x',x)}_{=0} + d(x,y') \leq d(x,y) + \underbrace{d(y,y')}_{=0}$$

Por tanto, $d(x,y) \leq d(x',y') \leq d(x,y)$, obteniendo la igualdad. Que $\hat{d}$ es una distancia es inmediato por la definición de la relación de equivalencia y las propiedades de $d$.

□

De los ejemplos anteriores podemos obtener los primeros espacios métricos a partir de cocientes:

- Si $(X,d)$ es un espacio métrico, la relación de equivalencia es la igualdad y el espacio cociente es esencialmente idéntico a $(X,d)$.
- Para cualquier conjunto $X$ no vacío, la pseudodistancia nula origina el espacio cociente de un solo punto, donde la aplicación nula sí es una distancia.
- Los espacios cociente de los $L^p$ bajo la relación dada por la pseudodistancia anterior (en este caso es la igualdad c.p.d.) son los conocidos espacios de Banach de funciones integrables $\mathcal{L}^p$

### 5.1.3. Distancias de Mahalanobis

Dentro de los espacios normados de dimensión finita nos encontramos con un conjunto de pseudodistancias muy útiles en el campo de la computación. Estas vienen dadas por matrices semidefinidas positivas, e independientemente de si se tratan de distancias propias o únicamente de pseudodistancias, se les conoce como distancias de Mahalanobis.

**Definición 5.1.3.** Sea $d \in \mathbb{N}$ y $M \in \mathcal{M}_d(\mathbb{R})_0^+$. La *distancia de Mahalanobis* asociada a la matriz $M$ es la aplicación $d_M \colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_0^+$, dada por

$$d_M(x,y) = \sqrt{(x-y)^T M (x-y)}.$$

Es claro que $d_M$ es una pseudodistancia, pues esta pseudodistancia procede de la seminorma $\|x\|_M = \sqrt{x^T M x}$, que a su vez procede del (pseudo-)producto escalar $\langle x,y\rangle_M = x^T M y$. Por tanto, cuando $M$ es definida positiva, $d_M$ da a $\mathbb{R}^d$ una estructura de espacio de Hilbert. Observemos que el espacio euclídeo usual está incluido en estos espacios, y se presenta cuando $M = I$.

En ocasiones se emplea el término *distancia de Mahalanobis* refiriéndose a la distancia al cuadrado, $d_M^2$. En el ámbito de la computación es mucho más eficiente trabajar con $d_M^2$ que con $d_M$, pues se evita así el cálculo de las raíces cuadradas. Aunque $d_M^2$ no sea realmente una distancia, mantiene las propiedades más útiles de $d_M$ desde el punto de vista de la computación, como por ejemplo, la mayor o menor cercanía entre distintas parejas de puntos. Por eso está bastante extendido el uso de *distancia de Mahalanobis* tanto para $d_M$ como para $d_M^2$.

Como hemos comentado, las distancias de Mahalanobis inducen (salvo aquellas que no verifican la propiedad de coincidencia) una estructura de espacio de Hilbert en $\mathbb{R}^d$. El recíproco también es cierto, puesto que toda forma bilineal simétrica en $\mathbb{R}^d$ viene determinada por una matriz simétrica. Si la forma es además semidefinida positiva (induciendo una pseudodistancia) también lo será la matriz, y si la

forma es definida positiva (siendo por tanto un producto escalar), la matriz asociada será también definida positiva. Por tanto, todos los productos escalares (incluyendo aquellos semidefinidos) vienen determinados por marices semidefinidas positivas.

Finalmente, al igual que las pseudodistancias inducían espacios métricos en el cociente, lo mismo ocurre con las distancias asociadas a matrices semidefinidas positivas singulares. En este caso dicho cociente implica una reducción de la dimensión del espacio, como vamos a ver a continuación. Esto es de nuevo una ventaja computacional, pues permitirá trabajar con datos de menor tamaño.

**Proposición 5.1.11.** *Sea $d_M$ una distancia de Mahalanobis en $\mathbb{R}^d$. Entonces,*

a) $V = \{x \in \mathbb{R}^d \colon d_M(x,0) = \|x\|_M = 0\}$ *es un subespacio vectorial de* $\mathbb{R}^d$.

b) *Si* $\dim(V) = m$, $\mathbb{R}^d/V$ *es un espacio vectorial de dimensión* $d' = d - m$, *que como conjunto cociente es igual a* $\mathbb{R}^d/\sim$ *($\sim$ es la relación de equivalencia del teorema 5.1.10)*

c) $d_M$ *induce en* $\mathbb{R}^d/V$ *una distancia de Mahalanobis* $d_{M'}$, *donde* $M' \in \mathcal{M}_{d'}(\mathbb{R})$ *es definida positiva.*

*Demostración.*

a) Sean $u, v \in V, \lambda, \mu \in \mathbb{R}$. Entonces,

$$0 \leq \|\lambda u + \mu v\|_M \leq \|\lambda u\|_M + \|\mu v\|_M = |\lambda|\|u\|_M + |\mu|\|v\|_M = 0,$$

luego $\|\lambda u + \mu v\|_M = 0$ y $\lambda u + \mu v \in V$.

b) Recordemos que la relación de equivalencia definida por un subespacio vectorial viene dada por $x \sim_V y \iff x - y \in V$, los elementos en el cociente son de la forma $V \in [x] = x + V := \{x + v \colon v \in V\}$, y la suma y producto por escalares vienen dados por $[x] + [y] = [x + y], a[x] = [ax]$, para $x, y \in \mathbb{R}^d$, $a \in \mathbb{R}$. La comprobación de que estas operaciones no dependen del representante es inmediata.

La aplicación $p \colon \mathbb{R}^d \to \mathbb{R}^d/V$ dada por $p(x) = [x]$ es lineal, gracias a la definición de las operaciones en el cociente, y sobreyectiva. Además,

$$\ker(p) = \{x \in \mathbb{R}^d \colon x + V = V\} = \{x \in \mathbb{R}^d \colon x \in V\} = V.$$

Por tanto,

$$\dim(\mathbb{R}^d/V) = \dim(\mathrm{im}(p)) = \dim(\mathbb{R}^d) - \dim(\ker(p)) = \dim(\mathbb{R}^d) - \dim(V) = d - m.$$

Finalmente, observamos que $\sim_V = \sim$ debido a la invarianza por traslaciones de las pseudodistancias asociadas a normas. Si $x, y \in \mathbb{R}^d$,

$$x \sim_V y \iff x - y \in V \iff d_M(x - y, 0) = 0 \iff d_M(x, y) = 0 \iff x \sim y.$$

c) Llamamos $U = \mathbb{R}^d/V$. Definimos la aplicación $\langle \cdot, \cdot \rangle \colon U \times U \to \mathbb{R}$ por $\langle x + V, y + V \rangle = x^T M y$. Veamos que $\langle \cdot, \cdot \rangle$ es un producto escalar.

- Está bien definida. Si $x + V = x' + V \in U$ e $y + V = y' + V \in U$, entonces $x - x' \in V, y - y' \in V$, luego $\|x - x'\|_M = \|y - y'\|_M = 0$. Observemos que

$$\begin{aligned}\langle x + V, y + V \rangle - \langle x' + V, y' + V \rangle &= x^T M y - (x')^T M y + (x')^T M y - (x')^T M (y') \\ &= (x - x')^T M y + (x')^T M (y - y').\end{aligned}$$

La deigualdad de Cauchy-Schwarz 2.3.13 obliga a que los últimos sumandos sean 0, puesto que $\|x - x'\|_M = \|y - y'\|_M = 0$. En consecuencia, $\langle x + V, y + V \rangle = \langle x' + V, y' + V \rangle$.

- Es lineal. Si $\lambda, \mu \in \mathbb{R}$ y $x+V, y+V, z+V \in U$,

$$\begin{aligned}\langle \lambda(x+V)+\mu(y+V), z+V\rangle &= \langle(\lambda x+\mu y+V), z+V\rangle = (\lambda x+\mu y)^T Mz \\ &= \lambda(x^T Mz)+\mu(y^T Mz) = \lambda\langle x+V, z+V\rangle + \mu\langle y+V, z+V\rangle.\end{aligned}$$

- Es simétrica. Si $x+V, y+V \in U$,

$$\langle x+V, y+V\rangle = x^T My = (x^T My)^T = y^T M^T x = y^T Mx = \langle y+V, x+V\rangle.$$

- Es definida positiva. Si $x+V \in U$, $\langle x+V, x+V\rangle = x^T Mx \geq 0$, y

$$\langle x+V, x+V\rangle = 0 \iff x^T Mx = 0 \iff \|x\|_M = 0 \iff x \in V \iff x+V = 0+V,$$

  luego $\langle x+V, x+V\rangle = 0$ si y solo si $x+V$ es el neutro en el espacio cociente, concluyendo que $\langle \cdot, \cdot \rangle$ es un producto escalar.

Por tanto, por ser un producto escalar, fijando una base ortonormal en $U$ y suponiendo los vectores en $U$ con coordenadas en dicha base, existe una matriz definida positiva $M' \in \mathcal{M}_{d'}(\mathbb{R})$ tal que $\langle x+V, y+V\rangle = (x+V)^T M'(y+V)$. Para concluir, basta ver que la distancia inducida por $M'$ es la distancia inducida por $d$ sobre el cociente. En efecto, si consideramos la distancia $\hat{d}\colon U \times U \to \mathbb{R}_0^+$ dada por el teorema 5.1.10, se tiene

$$\hat{d}(x+V, y+V)^2 = d(x,y) = (x-y)^T M(x-y) = \langle (x-y)+V, (x-y)+V\rangle = d^2_{M'}(x+V, y+V).$$

□

## 5.2. Descripción del problema

Uno de los componentes más importantes en muchos procesos cognitivos del ser humano consiste en la capacidad para detectar parecidos o semejanzas entre distintos objetos. Esta capacidad se ha llevado al campo del aprendizaje automático mediante el diseño de algoritmos que aprenden de un conjunto de datos de acuerdo con las similaridades entre dichos datos.

Para medir la similaridad entre los datos, es necesario introducir una distancia, la cual nos permite establecer una medida de cercanía mediante la cual se puede determinar cuándo un par de puntos es más similar que otro par de puntos. Sin embargo, como hemos visto en la sección anterior, existe una infinidad de distancias con las que podemos trabajar, y es posible que no todas ellas se adapten correctamente a nuestros datos, y no sean adecuadas para detectar las semejanzas entre estos. Por eso, la elección de una distancia adecuada es un elemento crucial en este tipo de algoritmos. La búsqueda de una distancia apropiada es la tarea que se lleva a cabo en el aprendizaje de métricas de distancia.

El *aprendizaje de métricas de distancia* (*DML*, *Distance Metric Learning*) es una disciplina del aprendizaje automático cuya finalidad es aprender distancias (incluyendo pseudodistancias, aunque no se indique explícitamente) a partir de un conjunto de datos. En su versión más general, se dispone de un conjunto de datos $\mathcal{X} = \{x_1, \ldots, x_N\}$, sobre el que se han recogido determinadas medidas de similitud entre distintos pares o tripletas de datos. Dichas similitudes vienen determinadas por los conjuntos

$$\begin{aligned}S &= \{(x_i, x_j) \in \mathcal{X} \times \mathcal{X}\colon x_i \text{ y } x_j \text{ son similares.}\} \\ D &= \{(x_i, x_j) \in \mathcal{X} \times \mathcal{X}\colon x_i \text{ y } x_j \text{ no son similares.}\} \\ R &= \{(x_i, x_j, x_l) \in \mathcal{X} \times \mathcal{X} \times \mathcal{X}\colon x_i \text{ es más similar a } x_j \text{ que a } x_l.\}.\end{aligned}$$

Con estos datos y conjuntos de similitud, el problema a resolver consiste en, fijada una familia de distancias $\mathcal{D}$, encontrar aquellas que mejor se adapten a los criterios especificados por los conjuntos de similitud. Para ello, se fija una determinada función de pérdida $\ell$, y las distancias buscadas serán aquellas que resuelvan el problema de optimización

$$\min_{d\in\mathcal{D}} \ell(d, S, D, R).$$

La selección de distintas funciones de pérdida es lo que conduce a las distintas técnicas de aprendizaje de métricas de distancia. Cada una de dichas funciones permitirá elaborar una determinada estrategia de optimización. Estos aspectos, para el caso del aprendizaje supervisado, se estudiarán en el siguiente capítulo.

A continuación vamos a concretar la descripción del problema. En primer lugar, si nos centramos en el aprendizaje supervisado, especialmente en el orientado a clasificación, además del conjunto $\mathcal{X}$ de datos, dispondremos de una lista de etiquetas $y_1, \ldots, y_N$ asociadas a cada dato. En este caso, la formulación general del problema se adapta fácilmente a la nueva situación, sin más que considerar los conjuntos $S$ y $D$ como

$$S = \{(x_i, x_j) \in \mathcal{X} \times \mathcal{X} \colon y_i = y_j\}$$
$$D = \{(x_i, x_j) \in \mathcal{X} \times \mathcal{X} \colon y_i \neq y_j\}$$

Adicionalmente, se puede disponer del conjunto $R$ definiendo tripletas $(x_i, x_j, x_l)$, donde en general $y_i = y_j \neq y_l$, verificándose además determinadas condiciones sobre la distancia entre $x_i$ y $x_j$, frente a la distancia entre $x_i$ y $x_l$. Este es el caso, por ejemplo, de los impostores en el algoritmo LMNN (véase la sección 6.2.1). En cualquier caso, las etiquetas disponen de toda la información necesaria en el ámbito del aprendizaje de métricas supervisado. En adelante nos centraremos en problemas de este tipo.

Por otro lado, centrándonos en la naturaleza del conjunto de datos, prácticamente la totalidad de la teoría del aprendizaje de métrica de distancias se desarrolla para datos numéricos, debido en parte a la riqueza de las distancias de las que disponen y a su facilidad para ser parametrizadas computacionalmente, y en parte a que los datos de naturaleza nominal pueden ser convertidos a variables numéricas binarias u ordinales con un preprocesamiento adecuado. Por ello, nos centraremos de ahora en adelante en problemas de aprendizaje supervisado con datos numéricos.

Supongamos entonces que $\mathcal{X} \subset \mathbb{R}^d$. Como vimos en la sección anterior, para espacios vectoriales de dimensión finita podemos tomar la familia de distancias de Mahalanobis, $\mathcal{D} = \{d_M \colon M \in \mathcal{M}_d(\mathbb{R})_0^+\}$. Con esta familia tenemos a nuestra disposición todas las distancias asociadas a productos escalares en $\mathbb{R}^d$ (y en menores dimensiones para el caso de las pseudodistancias), y viene determinada por el conjunto de las matrices semidefinidas positivas, y por ello, podemos utilizar estas matrices para parametrizar las distancias. De esta forma, el problema general adaptado al aprendizaje supervisado con distancias de Mahalanobis podemos reescribirlo como

$$\min_{M\in\mathcal{M}_d(\mathbb{R})_0^+} \ell(d_M, (x_1, y_1), \ldots, (x_N, y_N)). \tag{5.1}$$

Sin embargo, esta no es la única forma de parametrizar este tipo de problemas. Sabemos, por el teorema 2.3.12 que si $M \in \mathcal{M}_d(\mathbb{R})_0^+$, existe una matriz $L \in \mathcal{M}_d(\mathbb{R})$ tal que $M = L^T L$ y dicha matriz es única salvo una isometría. Entonces se tiene que

$$d_M^2(x, y) = (x-y)^T M(x-y) = (x-y)^T L^T L(x-y) = (L(x-y))^T (L(x-y)) = \|L(x-y)\|_2^2.$$

Por tanto, podemos parametrizar las distancias también mediante cualquier matriz, aunque en este caso la interpretación es distinta. Cuando aprendemos distancias mediante matrices semidefinidas positivas estamos aprendiendo una nueva métrica sobre $\mathbb{R}^d$. Cuando los aprendemos mediante las matrices $L$ anteriores, estamos aprendiendo una aplicación lineal que transforma los datos en el espacio, y la distancia asociada es la distancia euclídea usual de los datos proyectados en el nuevo espacio. Ambos enfoques son equivalentes gracias al teorema 2.3.12.

En cuanto a la dimensionalidad, es importante destacar que cuando la métrica aprendida $M$ no tiene rango máximo, realmente estamos aprendiendo una distancia sobre un espacio de dimensión inferior (por la proposición 5.1.11), lo que nos permite reducir la dimensionalidad de nuestro conjunto de datos. Lo mismo ocurre cuando aprendemos aplicaciones lineales $L$ que no tienen rango máximo. Podemos extender este caso y optar por aprender matrices $L \in M_{d' \times d}(\mathbb{R})$, con $d' < d$. De esta forma, aseguramos que los datos se proyectan directamente a un espacio de dimensión no superior a $d'$.

Ambos enfoques, tanto el de aprender la métrica $M$ como el de aprender la transformación $L$, son de gran utilidad para parametrizar los problemas del aprendizaje de métricas de distancia, cada uno con sus ventajas e inconvenientes. Por ejemplo, las parametrizaciones a través de $M$ suelen conducir a problemas de optimización convexos. En cambio, la convexidad en los problemas parametrizados por $L$ no es tan fácil de conseguir. Por otra parte, las parametrizaciones a través de $L$ permiten aprender directamente proyecciones a espacios de menor dimensión, mientras que las restricciones de dimensión para los problemas parametrizados por $M$ no son fáciles de satisfacer. Veamos estas diferencias con ejemplos sencillos.

Ejemplo 5.2.1: Muchas de las funciones que querremos optimizar dependerán de la distancia al cuadrado definida por la métrica $M$ o por la transformación $L$, es decir, o bien tendrán términos de la forma $\|v\|_M^2 = v^T M v$ o bien de la forma $\|v\|_L^2 = \|Lv\|_2^2$. Tanto la aplicación $M \mapsto \|v\|_M^2$ como la aplicación $L \mapsto \|v\|_L^2$ son convexas (la primera es, de hecho afín). Sin embargo, si queremos restar términos de esta forma, perdemos la convexidad en $L$, pues la aplicación $L \mapsto -\|v\|_L^2$ no es convexa. En cambio, la aplicación $M \mapsto -\|v\|_M^2$ sigue siendo afín y, por tanto, convexa.

Ejemplo 5.2.2: Las restricciones de rango no son convexas, y por tanto no disponemos de una proyección convexa sobre dicha restricción para hacer cumplir dichas restricciones durante el proceso de aprendizaje, salvo que aprendamos directamente la proyección (parametrizada por $L$) al espacio con la dimensión deseada. Por ejemplo, si consideramos el conjunto $C = \{M \in \mathcal{M}_2(\mathbb{R})_0^+ : r(A) \leq 1\}$, se tiene que $A = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} \in C$ y $B = \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix} \in C$. Sin embargo, $(1-\lambda)A + \lambda B = I \notin C$, para $\lambda = 1/2$.

## 5.3. Aplicaciones

En esta sección se describen algunas de las principales aplicaciones del aprendizaje de métricas de distancia, ilustradas con algunos ejemplos.

- **Mejorar la actuación de clasificadores basados en distancias.** Esta es una de las principales finalidades del aprendizaje de métricas supervisado. Mediante dicho aprendizaje, se encuentra una distancia que se adapte bien a los datos y al clasificador, mejorando el rendimiento de este último. En la figura 5.1 se muestra un ejemplo.
- **Reducción de la dimensionalidad.** Como ya hemos comentado, aprender una métrica de rango no máximo implica una reducción de dimensionalidad sobre los datos con los que trabajamos. Dicha reducción de dimensionalidad proporciona numerosas ventajas, como la reducción del coste

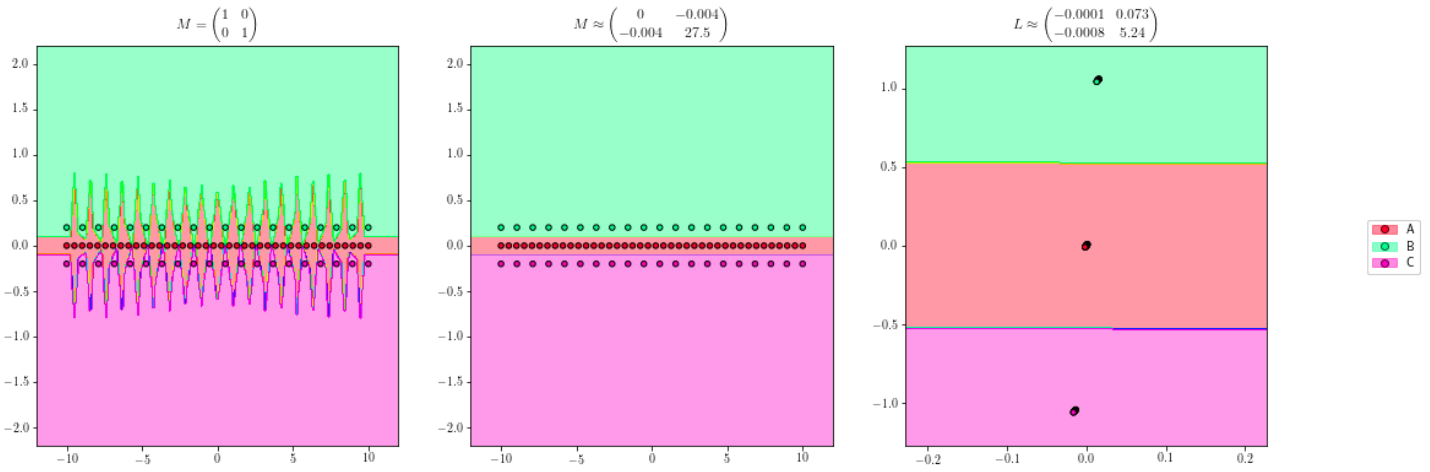


Figura 5.1: Supongamos que tenemos un conjunto de datos en el plano, los cuales pueden pertenecer a tres clases distintas, cuyas regiones vienen definidas por rectas paralelas. Supongamos que para clasificar un nuevo dato lo hacemos asignándole la clase del punto que se encuentre más cerca, para la distancia euclídea usual. Entonces, para los datos observados obtendríamos unas regiones de clasificación como las de la figura de la izquierda, pues los datos de las clases B y C están mucho más separados entre sí que la separación entre las regiones. Sin embargo, si aprendemos una distancia adecuada y volvemos a intentar clasificar asignando la clase del punto más cercano para esta nueva distancia, obtenemos unas regiones de clasificación como las de la figura central, mucho más efectivas. Por último, aprender una métrica es equivalente a aprender una transformación de los datos y medir en el espacio transformado con la distancia euclídea usual. Esto se muestra en la figura derecha. También podemos observar que los datos se están proyectando, salvo errores de precisión, sobre una recta, luego también estamos reduciendo la dimensionalidad del conjunto de datos.

computacional, tanto en espacio como en tiempo, de los algoritmos que se utilizarán posteriormente, o la eliminación del posible ruido introducido al tomar los datos. También, como veremos en la próxima sección, algunos algoritmos basados en distancias están expuestos a un problema denominado *maldición de la dimensionalidad*. Reduciendo la dimensión de los datos, dicho problema también se hace menos grave. Por último, si se estima necesario, las proyecciones a dimensión 1,2 y 3 nos permitirían obtener representaciones visuales de nuestros datos, como se muestra en la figura 5.2

- **Cambio de ejes y reorganización de los datos.** Muy relacionada con la reducción de dimensionalidad, esta aplicación se debe a aquellos algoritmos que aprenden transformaciones que permiten mover (o seleccionar según la dimensión) los ejes de coordenadas, de forma que en el nuevo sistema de coordenadas los vectores concentren determinadas medidas de información en sus primeras componentes. Un ejemplo se muestra en la figura 5.3.

- **Mejorar la actuación de los algoritmos de clustering.** Muchos de los algoritmos de clustering utilizan una distancia para medir la cercanía entre los datos, y así establecer los agrupamientos de forma que los datos presentes en un mismo grupo son cercanos para dicha distancia. En ocasiones, aunque desconozcamos los agrupamientos ideales de los datos ni el número de clusters a establecer, sí podemos saber que determinados pares de puntos deben estar en un mismo cluster y que otros determinados pares deben estar en clusters distintos. Esto ocurre en numerosos problemas, como por ejemplo, en el agrupamiento de documentos web. Dichos documentos poseen gran cantidad de información adicional, como es el caso de los links entre documentos, la cual nos puede incluirse como restricciones de similitud.

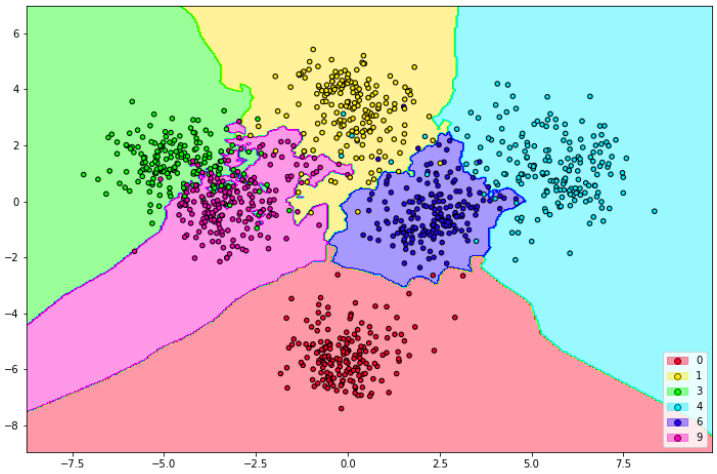


Figura 5.2: El dataset 'Dígitos' está formado por 1797 ejemplos. Cada uno de ellos consiste en un vector de 64 atributos, representando valores de intensidad sobre una imagen 8x8. Los ejemplos pertenecen a 10 clases distintas, cada una de ellas representando los números del 0 al 9. Aprendiendo una transformación adecuada somos capaces de proyectar la mayoría de clases sobre el plano, de forma que se perciban regiones claramente diferenciadas asociadas a cada una de las clases.

- **Aprendizaje semisupervisado.** El aprendizaje semisupervisado es un modelo de aprendizaje en el que se dispone de un conjunto de datos etiquetados y otro conjunto (en general mucho más grande) de datos sin etiquetar. Con ambos conjuntos de datos se busca aprender un modelo que permita etiquetar nuevos datos. El aprendizaje semisupervisado surge debido a que en muchas ocasiones la recopilación de datos sin etiquetar es relativamente sencilla, pero la asignación de etiquetas puede requerir que un supervisor las tenga que asignar manualmente, lo que puede ser inviable. En cambio, cuando se utilizan muchos datos no etiquetados junto con una pequeña cantidad de datos etiquetados es posible mejorar considerablemente los resultados del aprendizaje, como se ejemplifica en la figura 5.4. Muchas de estas técnicas consisten en construir un grafo con aristas ponderadas a partir de los datos, donde el valor de las aristas depende de las distancias entre los datos. A partir de dicho grafo se trata de inferir las etiquetas de todo el conjunto de datos, mediante distintos algoritmos de propagación [32, 42]. En la construcción del grafo la elección de una distancia adecuada es importante, entrando así en juego el aprendizaje de métricas de distancia.

Los algoritmos de aprendizaje supervisado analizados en este trabajo se centran en las tres primeras aplicaciones de la enumeración anterior.

## 5.4. El aprendizaje por semejanza

### 5.4.1. Introducción

El aprendizaje por semejanza es una disciplina del aprendizaje automático cuya finalidad es aprender a partir de la similitud con los datos en el conjunto de entrenamiento. De nuevo la similitud vendrá determinada por una función de distancia, por lo que el aprendizaje de una métrica apropiada previa a

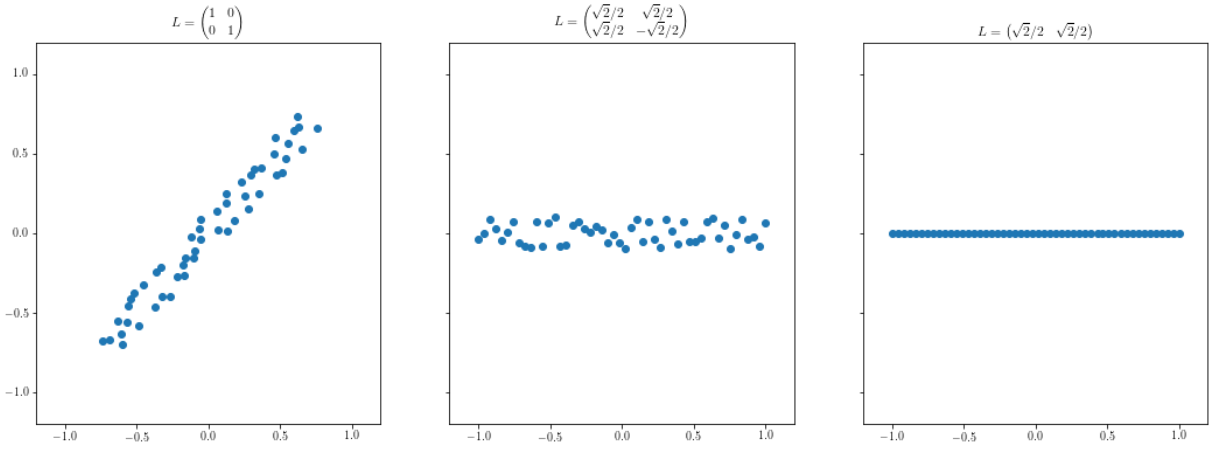


Figura 5.3: El conjunto de datos de la figura izquierda parece que concentra la mayoría de su información en la recta diagonal que une las esquinas inferior izquierda y la superior derecha. Aprendiendo la transformación adecuada, podemos conseguir que dicha dirección caiga sobre el eje horizontal, como se muestra en la figura central. De esta forma, la primera coordenada de los vectores en esta nueva base concentra gran parte de la variabilidad del vector. Además, parece razonable pensar que los valores que introduce la coordenada vertical pueden deberse a ruido, por lo que podemos incluso quedarnos únicamente con la primera componente, como se muestra en la figura derecha.

este proceso de aprendizaje aumentará la eficacia de este tipo de técnicas. De hecho, el aprendizaje por semejanza es una de las principales aplicaciones del aprendizaje de métricas de distancia, como se mostró en el primer ejemplo de la sección 5.3.

En el aprendizaje supervisado, el enfoque que sigue este paradigma es el que se muestra a continuación. Supongamos que queremos aprender un clasificador para un conjunto de datos que se distribuyen de acuerdo a una distribución $\mathcal{D}$ de probabilidad. Disponemos para ello de una muestra $(x_1, y_1), \ldots, (x_N, y_N)$ de datos etiquetados mediante una función de etiquetado desconocida $f\colon \mathcal{X} \to \mathcal{Y}$. Suponemos también que en el espacio $\mathcal{X}$ disponemos de una distancia $d$. Para un nuevo dato $x \sim \mathcal{D}$, le asignamos su clase a través de una función $h\colon \mathcal{X} \to \mathcal{Y}$ dada por $h(x) = \phi(d, x, (x_1, y_1), \ldots, (x_N, y_N))$, una función que depende únicamente de los datos y de la distancia entre ellos.

Notemos que en este tipo de aprendizaje no buscamos un clasificador $h$ en una familia de hipótesis $\mathcal{H}$, de forma que se minimice una determinada función, sino que partimos de una función $h$ prefijada. Esto hace que el proceso de aprendizaje propiamente dicho consista únicamente en almacenar los datos en memoria, mientras que el esfuerzo computacional se realiza durante el proceso de predicción, en el que se evalúa la función de distancia. Este tipo de clasificadores, en los que el proceso de aprendizaje o generalización se retrasa hasta el momento en el que se desea predecir un nuevo dato, se denominan *clasificadores perezosos*.

Por último, es interesante observar cómo el aprendizaje de métricas de distancia complementa a este tipo de clasificadores perezosos. Como ya hemos dicho, las técnicas de aprendizaje por semejanza no tienen un proceso de aprendizaje propiamente dicho y parten de una función hipótesis predefinida. En cambio, el aprendizaje de métricas de distancia sí parte de un conjunto de hipótesis, en concreto, el conjunto de distancias de Mahalanobis, como se mostraba en la expresión 5.1. Podemos combinar ambas técnicas obteniendo así un clasificador por semejanza que durante el aprendizaje encuentra una función hipótesis, dependiente de una distancia, que minimiza una función de pérdida definida para el

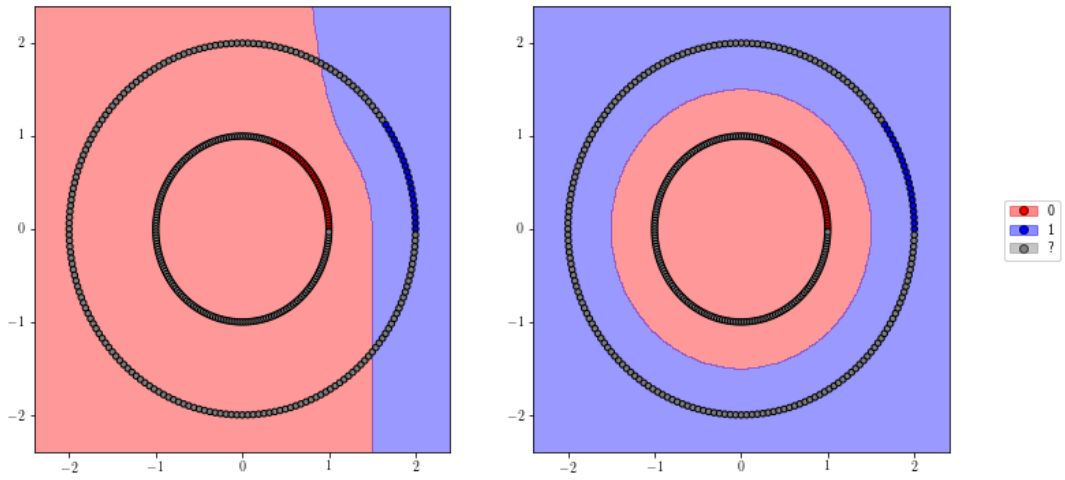


Figura 5.4: Aprendizaje con la información supervisada (izquierda) frente al aprendizaje considerando toda la información no supervisada (derecha).

conjunto de distancias de Mahalanobis.

El clasificador por semejanza más popular es el de vecinos cercanos, que analizaremos en la siguiente sección.

### 5.4.2. El clasificador de vecinos cercanos.

El clasificador de vecinos cercanos es un clasificador por semejanza muy conocido y que, como su propio nombre indica, clasifica los nuevos datos de acuerdo con la clase de sus vecinos más cercanos. Supongamos que tenemos la muestra de entrenamiento $(x_1, y_1), \ldots, (x_N, y_N)$ y un dato a predecir, $x \in \mathcal{X}$. Definimos, para dicho $x$, una permutación $(\pi_1(x), \ldots, \pi_N(x))$ del conjunto $\{1, \ldots, N\}$ de forma que los datos de entrenamiento quedan ordenados por dicha permutación según su distancia a $x$, es decir, se tiene que

$$d(x, x_{\pi_i}(x)) \leq d(x, x_{\pi_{i+1}}(x)) \quad i = 1, \ldots, N-1.$$

Entonces, el clasificador de los $k$ vecinos cercanos o k-NN asigna a $x$ el valor más repetido en la lista $(y_{\pi_1(x)}, \ldots, y_{\pi_k(x)})$, es decir, la clase mayoritaria de sus $k$ vecinos más cercanos. Cuando $k = 1$, la función hipótesis $h$ viene dada por $h(x) = y_{\pi_1}(x)$. En este caso, las regiones que determinan cada posible vecino más cercano vienen determinadas por politopos convexos (la generalización de polígonos y poliedros) y se denominan celdas de Voronoi. En el caso bidimensional, las regiones se pueden visualizar mediante diagramas de Voronoi (figura 5.5).

Es interesante destacar que, por cómo se asignan las clases para los nuevos datos, el k-NN permite trabajar en problemas de clasificación multiclase sin ningún tipo de limitación. Esto, normalmente, no es así para muchos clasificadores, los cuales están diseñados únicamente para resolver problemas de clasificación binarios. Para extender estos clasificadores a problemas multiclase, la estrategia más común es dividir el problema en subproblemas binarios, resolver estos problemas, y asignar la clase final

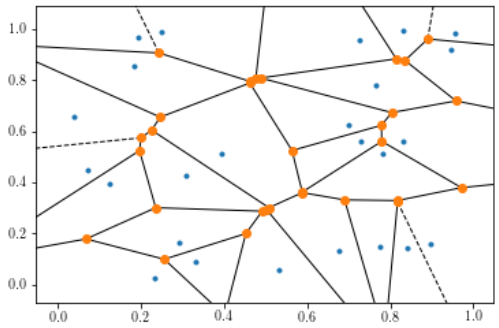


Figura 5.5: Diagrama de Voronoi.

como aquella mayoritaria obtenida en los subproblemas. Los métodos usuales de división en problemas binarios consisten en enfrentar una clase frente a todas las demás (*One versus All*), o enfrentar todos los pares de clases entre sí (*One versus One*) [13]. En general, la mayoría de técnicas que aprenden por semejanza permiten también trabajar directamente con problemas multiclase.

La elección del número de vecinos $k$ puede influir bastante en la región delimitada por el clasificador. Dicha región es no paramétrica, al no hacerse ninguna asunción sobre la forma de la función hipótesis $h$. Los valores pequeños de $k$ se ajustan más a los datos de entrenamiento, generando así una región más puntiaguda. En tales casos, el sesgo es bajo pero la variabilidad es alta. Los papeles se intercambian para valores grandes de $k$, donde la región generada presenta un aspecto más suave, como se muestra en la figura 5.6.

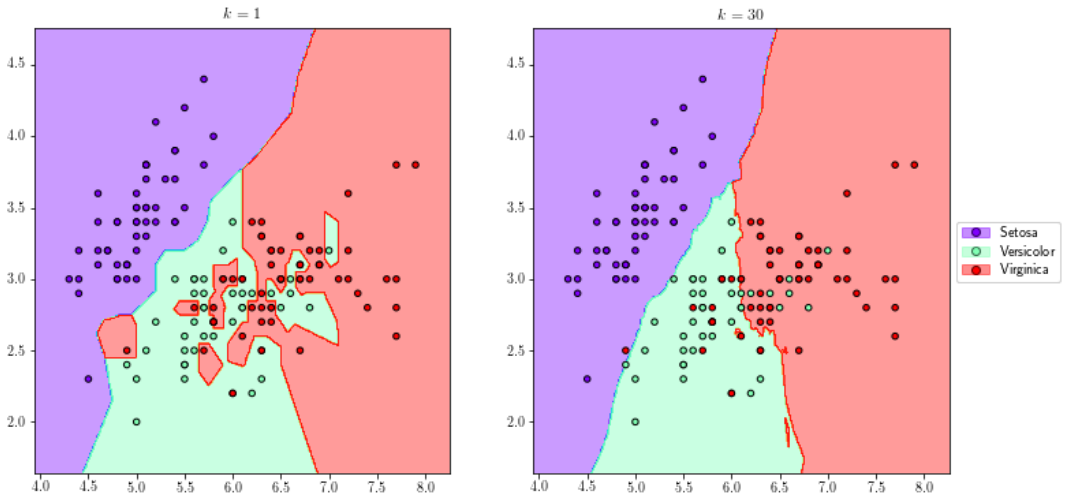


Figura 5.6: Comparación del k-NN para distintos valores de $k$ (dataset 'Iris')

Como ya se anticipó en la sección, el k-NN sufre de la conocida como *maldición de la dimensionalidad*. Nos restringimos al caso $k = 1$. Supongamos que el dominio del problema es $\mathcal{X} = [0, 1]^d$ (es decir,

trabajamos con datos normalizados) y suponemos que el conjunto de clases es $\mathcal{Y} = \{0, 1\}$. Suponemos que $\mathcal{D}$ es una distribución sobre $\mathcal{X} \times \mathcal{Y}$ para la cual la distribución de probabilidad condicionada, $\eta(x) = \mathbb{P}[y = 1|x]$ es $c$-lipschitziana para la distancia $d$ con la que trabajamos. Entonces, es posible probar que la diferencia entre el error empírico y el *error bayesiano óptimo*, asociado a la hipótesis $h^*(x) = \mathbb{1}_{[\eta(x)>1/2]}$ puede acotarse por $4c\sqrt{d}N^{-\frac{1}{d+1}}$ [31], donde $N$ es el tamaño del conjunto de entrenamiento. Por tanto, si queremos asegurar que dicho error sea menor que $\varepsilon$, necesitamos escoger $N \geq (4c\sqrt{d}/\varepsilon)^{d+1}$. Es decir, según esta cota, el número de ejemplos crece exponencialmente con la dimensionaldad del conjunto. Además, es posible encontrar distribuciones para las cuales esta cota sea necesaria, además de suficiente. Esta circunstancia se puede generalizar para cualquier $k$, y es lo que se conoce como la maldición de la dimensionalidad para el k-NN.

Por último, es importante comentar que se puede cambiar la regla de clasificación del k-NN. Por ejemplo para, en lugar de asignar la clase mayoritaria en los $k$ vecinos más cercanos, establecer una ponderación sobre cada vecino de forma inversamente proporcional a la distancia, teniendo así más peso las clases de los primeros vecinos más cercanos. También se puede extender esta regla a problemas de regresión, considerando por ejemplo la media, o una media ponderada, entre los $k$ vecinos más cercanos. En general, un clasificador o regresor de $k$ vecinos cercanos utiliza una función hipótesis que se puede expresar genéricamente como

$$h(x) = \phi(x, d, (x_{\pi_1(x)}, y_{\pi_1(x)}), \ldots, (x_{\pi_N(x)}, y_{\pi_N(x)})).$$

### 5.4.3. Otros clasificadores por semejanza.

Aunque el k-NN es el algoritmo por semejanza más popular para clasificación, no es el único. A continuación se muestran otros clasificadores relevantes.

- **El clasificador de la media más cercana.** Denominado NCM (*Nearest Class Mean*), este clasificador, durante el proceso de aprendizaje, calcula los vectores media de cada clase. Después, a la hora de predecir un nuevo dato, le asigna la clase del vector media más cercano. Es un clasificador muy eficiente y simple, aunque su simplicidad lo convierte en un clasificador bastante débil frente a conjuntos que no se agrupan en torno a su media. Existe la posibilidad de generalizarlo a múltiples centroides, como veremos en la sección 6.3.2.
- **El clasificador de vecinos cercanos por radio**. Este clasificador es muy similar al k-NN, solo que en este caso, en vez de fijar un número de vecinos $k$, se fija un radio $R$. A la hora de clasificar un nuevo dato $x$, se buscan todos los datos del conjunto de entrenamiento que disten de $x$ menos que $R$. Todos los datos encontrados serán los vecinos cercanos de $x$, y a $x$ se le asignará la clase mayoritaria entre dichos vecinos. Notemos que en este caso, el número de vecinos varía con cada ejemplo, pudiendo incluso no haber vecinos. En este caso, la elección de un radio adecuado es muy importante, y puede presentar un comportamiento inadecuado en conjuntos de datos con grandes variaciones de densidad (podría haber zonas en las que apenas hay vecinos y zonas con un número elevado de vecinos).

# Capítulo 6

# Descripción teórica de técnicas de aprendizaje de métricas de distancia

En este capítulo se describen algunas de las técnicas más populares actualmente en el aprendizaje de métricas de distancia supervisado. A ellas se añade el análisis de componentes principales, pese a no ser supervisado, debido a su importancia para otros algoritmos de aprendizaje de métricas. Algunas estas técnicas, como PCA o LDA, constituyen procedimientos estadísticos desarrollados a finales del siglo pasado, que en la actualidad siguen siendo de gran relevancia en muchos problemas. Otras propuestas más recientes se sitúan en el estado del arte, como es el caso de NCMML o DMLMJ, entre otras.

Las técnicas analizadas se agrupan en seis secciones. Cada una de estas secciones describe algoritmos que comparten una misma finalidad principal, si bien las finalidades que describen cada sección no son exclusivas. En la primera sección se estudian las técnicas orientadas específicamente a la reducción de dimensionalidad. A continuación, se desarrollan las técnicas cuya finalidad es aprender distancias que mejoren el clasificador kNN, seguidas de aquellas que buscan mejorar los clasificadores basados en centroides. La cuarta sección incluye los métodos basados en la teoría de la información, especialmente en las divergencias, aprendiendo así distancias que acerquen o alejen determinadas distribuciones de probabilidad según la divergencia medida. Posteriormente se describen varios mecanismos de aprendizaje de distancias con objetivos menos específicos. Por último, se analizan las versiones basadas en kernels de algunos de los algoritmos anteriores, para trabajar en espacios de alta dimensionalidad.

Para cada una de las técnicas se analizará el problema que buscan resolver u optimizar, las formulaciones matemáticas de dichos problemas y los algoritmos propuestos para resolverlos.

## 6.1. Técnicas de reducción de dimensionalidad

### 6.1.1. PCA

El *análisis de componentes principales* (PCA, *Principal Component Analysis*) es una de las técnicas más populares de reducción de dimensionalidad en el ámbito del aprendizaje de métricas de distancia. Aunque se trata de una técnica de aprendizaje sin ningún tipo de supervisión, resulta necesario hablar de ella en este trabajo, por un lado por su gran relevancia, y más en particular, porque PCA es la herramienta de reducción de dimensionalidad por excelencia utilizada en los algoritmos de aprendizaje de distancias supervisados que no admiten de por sí una reducción de dimensionalidad. En tales algoritmos, PCA se aplica primeramente sobre los datos para poder utilizar posteriormente el algoritmo en el espacio de dimensión reducida.

El análisis de componentes principales puede entenderse desde dos puntos de vista diferentes, que

acaban conduciendo al mismo problema de optimización. El primero de estos enfoques consiste en encontrar dos transformaciones, una que comprima los datos a un espacio de menor dimensión, y otra que los descomprima en el espacio original, de forma que en el proceso de compresión y descompresión se pierda la mínima información.

Vamos a centrarnos en este primer enfoque. Supongamos que tenemos el conjunto de datos $\mathcal{X} = \{x_1, \dots, x_N\} \subset \mathbb{R}^d$, y fijamos $d' < d$. Vamos a suponer además que los datos están centrados, es decir, que su media es cero. Si no lo fuera, basta con aplicar previamente a los datos la transformación $x \mapsto x - \mu$, donde $\mu = \sum x_i/N$ es la media de los datos. Buscamos una matriz de compresión $L \in \mathcal{M}_{d' \times d}(\mathbb{R})$ y una matriz de descompresión $U \in \mathcal{M}_{d \times d'}(\mathbb{R})$ de forma que, tras comprimir y descomprimir cada dato, los cuadrados de las distancias euclídeas al dato original sean mínimos. Es decir, el problema que buscamos resolver es

$$\min_{\substack{L \in \mathcal{M}_{d' \times d}(\mathbb{R}) \\ U \in \mathcal{M}_{d \times d'}(\mathbb{R})}} \quad \sum_{i=1}^{N} \|x_i - ULx_i\|_2^2. \tag{6.1}$$

Para encontrar una solución a este problema, en primer lugar vamos a ver que las matrices $U$ y $L$ han de estar relacionadas de una forma muy particular.

**Lema 6.1.1.** *Si $(U, L)$ es una solución del problema 6.1, entonces $LL^T = I$ (en $\mathbb{R}^{d'}$) y $U = L^T$.*

*Demostración.* Fijamos $U \in \mathcal{M}_{d \times d'}(\mathbb{R})$ y $L \in \mathcal{M}_{d' \times d}(\mathbb{R})$. Podemos suponer que tanto $U$ como $L$ tienen rango máximo, pues en caso contrario el rango de $UL$ es menor que $d'$. Notemos que en tal caso, siempre es posible extender las matrices $U$ y $L$ a rango máximo de forma que el subespacio generado extienda al generado por $UL$ (basta sustituir los vectores linealmente dependientes en las columnas de la matriz por vectores independientes mientras la dimensión lo permita), y en tal caso el error obtenido en 6.1 para la extensión va a ser, a lo sumo, el error obtenido para $U$ y $L$.

Consideramos la aplicación $x \mapsto ULx$. La imagen de esta aplicación, $R = \{ULx \colon x \in \mathbb{R}^d\}$, es un subespacio vectorial de $\mathbb{R}^d$ de dimensión $d'$. Sea $\{u_1, \dots, u_{d'}\}$ una base ortonormal de $R$, y sea $V \in \mathcal{M}_{d' \times d}(\mathbb{R})$ la matriz que tiene, por filas, los vectores $u_1, \dots, u_{d'}$. Se verifica entonces que la imagen de $V$ tiene rango $d'$ y que $VV^T = I$. Además, si consideramos $V^T$ como aplicación lineal, se tiene que su imagen es $R$ (puesto que $V^T e_i = u_i, i = 1, \dots, d'$, donde $\{e_1, \dots, e_{d'}\}$ es la base usual de $\mathbb{R}^{d'}$).

Por tanto, todos los vectores de $R$ pueden escribirse como $V^T y$, con $y \in \mathbb{R}^{d'}$. Dados $x \in \mathbb{R}^d, y \in \mathbb{R}^{d'}$, se tiene

$$\begin{aligned} \|x - V^T y\|_2^2 &= \langle x - V^T y, x - V^T y \rangle \\ &= \|x\|^2 - 2\langle x, V^T y \rangle + \|V^T y\|^2 \\ &= \|x\|^2 - 2\langle y, Vx \rangle + y^T V V^T y \\ &= \|x\|^2 - 2\langle y, Vx \rangle + y^T y \\ &= \|x\|^2 + \|y\|^2 - 2\langle y, Vx \rangle. \end{aligned}$$

Si calculamos el gradiente respecto de $y$ a partir de la última expresión anterior, obtenemos $\nabla_y \|x - V^T y\|_2^2 = 2y - 2Vx$, que, al igualar a cero, nos permite obtener un único punto crítico, $y = Vx$. La convexidad de esta función (es la composición de la norma euclídea con una aplicación afín) nos asegura que este punto crítico es un mínimo global. Por tanto, esto nos indica que, para cada $x \in \mathbb{R}^d$, la distancia a $x$ en el conjunto $R$ alcanza su mínimo en el punto $V^T V x$. En particular, para los datos del

conjunto $\mathcal{X}$ concluimos que

$$\sum_{i=1}^{N} \|x_i - ULx_i\|_2^2 \geq \sum_{i=1}^{N} \|x_i - V^T V x_i\|_2^2.$$

Podemos encontrar una matriz $V$ con estas propiedades para cualesquiera $U$ y $L$ en las condiciones del problema, lo que concluye la prueba. □

El lema anterior nos permite reformular nuestro problema en términos únicamente de la matriz $L$,

$$\min_{\substack{L \in \mathcal{M}_{d' \times d}(\mathbb{R}) \\ LL^T = I}} \quad \sum_{i=1}^{N} \|x_i - L^T L x_i\|_2^2. \tag{6.2}$$

Notemos ahora que, para $x \in \mathbb{R}^d$ y $L \in \mathcal{M}_{d' \times d}(\mathbb{R})$ con $LL^T = I$, se verifica

$$\begin{aligned} \|x - L^T Lx\|_2^2 &= \langle x - L^T Lx, x - L^T Lx \rangle \\ &= \|x\|^2 - 2\langle x, L^T Lx \rangle + \langle L^T Lx, L^T Lx \rangle \\ &= \|x\|^2 - 2x^T L^T Lx + x^T L^T LL^T Lx \\ &= \|x\|^2 - x^T L^T Lx \\ &= \|x\|^2 - \operatorname{tr}(x^T L^T Lx) \\ &= \|x\|^2 - \operatorname{tr}(Lxx^T L^T). \end{aligned}$$

Por tanto, si eliminamos los términos que no dependen de $L$, podemos transformar el problema 6.2 en el siguiente problema equivalente:

$$\max_{\substack{L \in \mathcal{M}_{d' \times d}(\mathbb{R}) \\ LL^T = I}} \quad \operatorname{tr}\left(L\Sigma L^T\right), \tag{6.3}$$

donde $\Sigma = \sum_{i=1}^{N} x_i x_i^T$ es, salvo una constante, la matriz de covarianza asociada a los datos de $\mathcal{X}$. Esta matriz es simétrica, y el teorema 2.4.6 garantiza que podemos encontrar un máximo del problema si construimos $L$ añadiendo los $d'$ vectores propios de $\Sigma$ correspondientes a sus $d'$ mayores valores propios. Estos vectores los podemos tomar ortonormales, por la simetría de $\Sigma$. Las direcciones que determinan estos vectores son las *direcciones principales*, y las componentes de los datos transformados en el sistema ortonormal determinado por las direcciones principales son las llamadas *componentes principales*.

Para concluir, el segundo enfoque desde el que se puede tratar el problema de los componentes principales consiste en la selección de las direcciones ortogonales para las que se maximice la varianza. Sabemos que si $\Sigma$ es la matriz de covarianza de $\mathcal{X}$, al aplicar una transformación $L$ a los datos la nueva matriz de covarianza viene dada por $L\Sigma L^T$. Si queremos una transformación que reduzca la dimensionalidad y para la cual se maximice la varianza en cada variable lo que buscamos es tomar la traza de la matriz anterior, lo que nos conduce de nuevo al problema 6.3. La simetría de $\Sigma$ garantiza que podamos tomar las direcciones principales ortonormales que maximicen la varianza para cada posible valor de $d'$.

Por último, es importante destacar que la matriz $L \in \mathcal{M}_d(\mathbb{R})$ (tomando todas las dimensiones) que se construye añadiendo por filas los vectores propios de $\Sigma$ es la matriz ortonormal que diagonaliza $\Sigma$, y

por tanto, al aplicar $L$ sobre los datos, los datos transformados tienen como matriz de covarianza la matriz diagonal $L\Sigma L^T = \text{diag}(\lambda_1, \ldots, \lambda_d)$, donde $\lambda_1, \ldots, \lambda_d$ son los valores propios de $\Sigma$. Esto nos dice que los valores propios de la matriz de covarianza representan la cantidad de varianza explicada por cada una de las direcciones principales. Esto proporciona una ventaja adicional al PCA, ya que permite analizar el porcentaje de varianza que explica cada componente principal para poder a posteriori elegir una dimensión que se ajuste a la cantidad de varianza que se quiera conservar en los datos transformados.

La figura 6.1 ejemplifica gráficamente el funcionamiento del análisis de componentes principales.

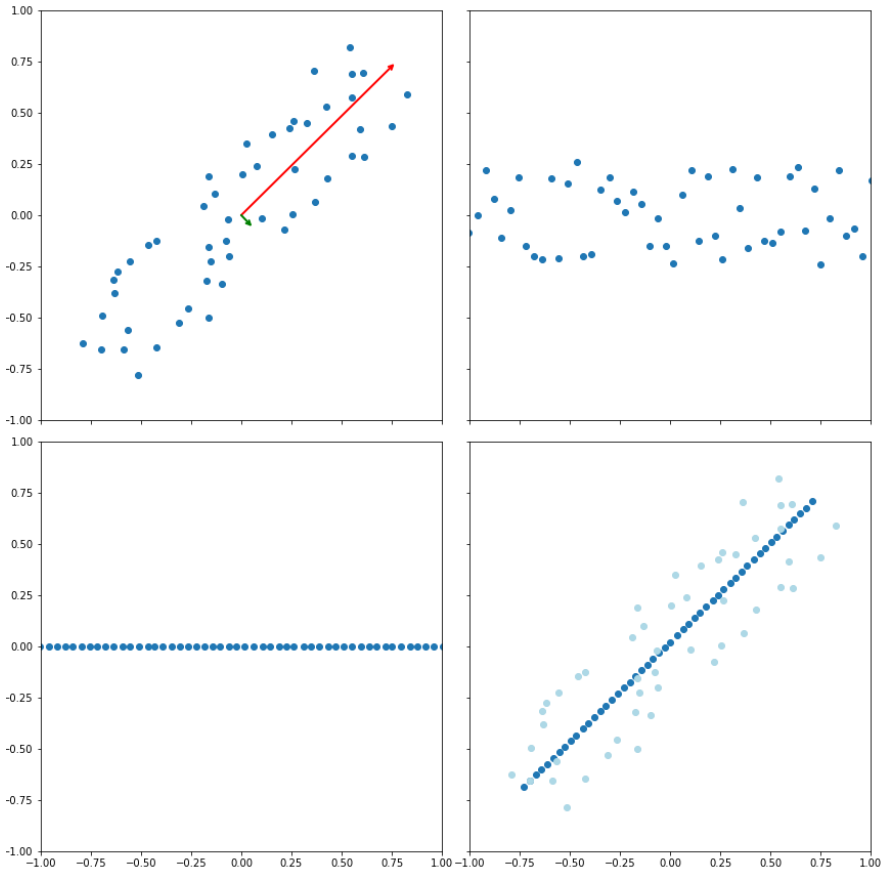


Figura 6.1: Ejemplificación gráfica del PCA. En la primera imagen se muestra un conjunto de datos, junto con las direcciones principales (proporcionales de acuerdo a la varianza explicada) aprendidas por PCA. A su derecha, los datos proyectados en dimensión máxima. Observamos que dicha proyección consiste en girar los datos haciendo coincidir los ejes con las direcciones principales. Abajo a la izquierda, los datos proyectados sobre la primera componente principal. Por último, a su derecha, los datos recuperados mediante la matriz de descompresión, junto con los datos originales. Podemos comprobar que la proyección de PCA es la que minimiza el error cuadrático de descompresión. En este caso particular los datos descomprimidos se encuentran en la recta de regresión de los datos originales, debido a las dimensiones del problema.

### 6.1.2. LDA

El análisis discriminante lineal (LDA, *Linear Discriminant Analysis*) es una técnica clásica de aprendizaje de métricas de distancia cuya finalidad es aprender una matriz de proyección que maximice la separación

entre clases en el espacio proyectado, es decir, trata de encontrar las direcciones que mejor permiten distinguir las distintas clases, como se muestra en la figura 6.2.

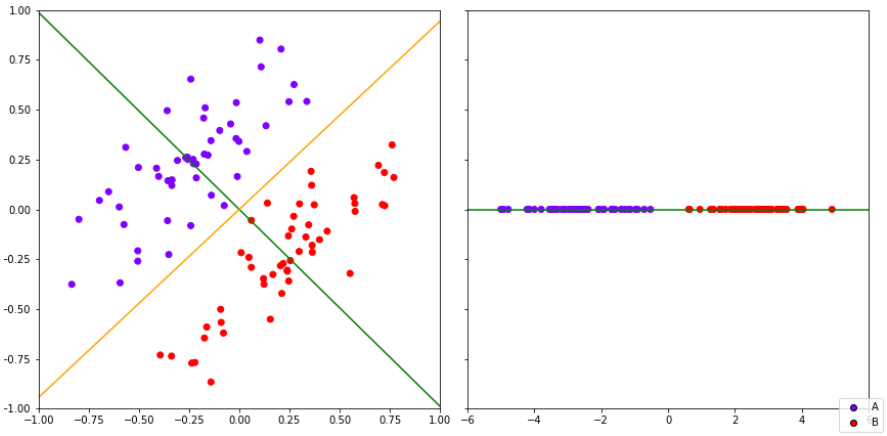


Figura 6.2: Ejemplo gráfico del LDA y comparación con PCA. En la primera imagen se muestra un conjunto de datos, con la dirección principal determinada por PCA, en naranja, y la dirección determinada por LDA, en verde. Observamos que si proyectamos los datos sobre la dirección obtenida por LDA quedan bien separados, como se muestra en la imagen derecha. En cambio, la dirección obtenida por PCA solo nos permite maximizar la varianza de todo el conjunto al proyectar, pues no considera la información de las etiquetas.

La figura 6.2 nos permite, además, comparar los resultados de las proyecciones obtenidas por PCA y LDA, mostrando la diferencia más notable entre ambas técnicas: PCA no tiene en cuenta la información de las clases, buscando las direcciones que maximizan la varianza del conjunto total de datos, mientras que LDA sí que utiliza la información presente en las etiquetas, obteniendo direcciones en las que mejor se pueden proyectar los datos para tener una buena separación de clases. Se puede apreciar que las direcciones obtenidas por PCA y LDA no presentan ningún tipo de relación, siendo esta última la única de las dos que proporciona una proyección de los datos orientada al aprendizaje supervisado.

También es posible observar en la figura 6.2 que no tiene sentido buscar una segunda dirección independiente que siga maximizando la separación de las clases, mientras que en PCA siempre tiene sentido ir buscando inductivamente direcciones ortogonales que maximicen la varianza. Si el conjunto de datos mostrado en la figura tuviera una tercera clase, podríamos encontrar una segunda dirección que maximizara la separación entre clases, ofreciendo así la posibilidad de proyectar sobre el plano. En general, vamos a ver que si tenemos $r$ clases podremos encontrar como mucho (y siempre que lo permita la dimensión del espacio original) $r-1$ direcciones que maximicen la separación. Esto nos indica que las proyecciones que va a aprender LDA van a ser, en general, hacia una dimensión bastante baja, y siempre limitada por el número de clases en el conjunto de datos.

Supongamos el conjunto de datos etiquetados $\mathcal{X} = \{x_1, \ldots, x_N\} \subset \mathbb{R}^d$ donde $\mathcal{C}$ es el conjunto de clases del problema, e $y_1, \ldots, y_N \in \mathcal{C}$ son las etiquetas asociadas a cada dato de $\mathcal{X}$. Supongamos que el número de clases del problema es $|\mathcal{C}| = r$. Para cada $c \in \mathcal{C}$ definimos el conjunto $\mathcal{C}_c = \{i \in \{1, \ldots, N\} \colon y_i = c\}$, y $N_c = |\mathcal{C}_c|$. Consideramos los vectores media de cada clase,

$$\mu_c = \frac{1}{N_c} \sum_{i \in \mathcal{C}_c} x_i,$$

y el vector media de todo el conjunto de datos,

$$\mu = \frac{1}{N}\sum_{c\in\mathcal{C}}\sum_{i\in\mathcal{C}_c} x_i = \frac{1}{N}\sum_{i=1}^{N} x_i.$$

Vamos a definir dos matrices de dispersion, una entre clases (denominada *between-class*), y otra entre los datos de mismas clases o intra-clase (denominada *within-class*), notadas como $S_b$ y $S_w$, respectivamente. La matriz de dispersión entre clases se define como

$$S_b = \sum_{c\in\mathcal{C}} N_c(\mu_c - \mu)(\mu_c - \mu)^T. \tag{6.4}$$

Y la matriz de dispersión intra-clase,

$$S_w = \sum_{c\in\mathcal{C}}\sum_{i\in\mathcal{C}_c} (x_i - \mu_c)(x_i - \mu_c)^T. \tag{6.5}$$

Notemos que estas matrices, representan, salvo constantes multiplicativas, las covarianzas entre los datos de las distintas clases tomando las medias como representantes de cada clase en el primer caso, y la suma, para cada clase, de las covarianzas para los datos de dicha clase, en el segundo caso. Como queremos maximizar la separación vamos a formular el problema de optimización como la búsqueda función de una proyección $L \in \mathcal{M}_{d'\times d}(\mathbb{R})$ que maximice el cociente entre las varianzas entre clase y las varianzas intra clase determinadas por las matrices anteriores. El problema se establece como

$$\max_{L\in\mathcal{M}_{d'\times d}(\mathbb{R})} \quad \operatorname{tr}\left((LS_wL^T)^{-1}(LS_bL^T)\right). \tag{6.6}$$

El teorema 2.4.8 nos asegura que para maximizar el problema 6.6 $L$ ha de estar compuesta por los vectores propios asociados a los valores propios de mayor valor de $S_w^{-1}S_b$, siempre que $S_w$ sea invertible. En la práctica, esto ocurre en la mayoría problemas donde $N \gg d$, pues $S_w$ es la suma de $N$ productos tensoriales, cada uno de los cuales puede aportar una dimensión al rango. Si $N \gg d$ es probable que $S_w$ tenga rango máximo. Esto, junto a que $S_w$ es semidefinida positiva garantizarían que $S_w$ fuera definida positiva, entrando así en las hipótesis del teorema.

Es interesante destacar el parecido entre el problema de optimización 6.6 y la expresión del índice de Calinski-Harabasz [22], un índice utilizado en clustering para medir la separación de las clases establecidas.

Por otra parte, notemos, como ya se adelantó al inicio de la sección, que a lo sumo podemos obtener $r-1$ vectores propios con valor propio asociado no nulo. Esto es debido a que $S_b$ tiene a lo sumo rango $r-1$, pues su rango coincide con el rango de la matriz $A$ que tiene por columnas los vectores $\mu_c - \mu$ (se verifica que $S_b = A\,\mathrm{diag}(N_{c_1},\dots,N_{c_r})A^T$), lo que da como rango a lo sumo $r$, y dicha matriz presenta además la combinación lineal $\sum N_c(\mu_c - \mu) = 0$. Por tanto, $S_w^{-1}S_b$ también tiene a lo sumo rango $r-1$. En consecuencia, la matriz de proyección que maximiza el problema 6.6 también va a tener, a lo sumo, este rango, luego la proyección va a quedar contenida en un espacio de dicha dimensión. Por tanto, la elección de una dimensión $d' > r-1$ no va a aportar ninguna información adicional a la que aporta la proyección en dimensión $r-1$.

Para concluir, aunque hemos visto que LDA permite reducir la dimensionalidad añadiendo información supervisada frente a la no supervisión de PCA, también puede presentar algunas limitaciones:

- Si la muestra de datos es demasiado pequeña, la matriz de dispersión intra-clase puede ser singular, impidiendo el cálculo de $S_w^{-1}S_b$. En esta situación, se han propuesto diversos mecanismos para seguir adelante con esta técnica. Uno de los más utilizados consiste en regularizar el problema, considerando, en lugar de $S_w$, la matriz $S_w + \varepsilon I$, donde $\varepsilon > 0$, haciendo que $S_w + \varepsilon I$ sea definida positiva. El problema de la singularidad de $S_w$ también surge si hay atributos correlacionados. Este caso se puede evitar eliminando atributos redundantes en un preprocesado previo al aprendizaje.
- La definición de las matrices de dispersión asume en cierta medida que los datos en cada clase se distribuyen mediante gaussianas multivariante. Por tanto, si los datos presentaran otras distribuciones, la proyección aprendida podría no ser de calidad.
- Como ya se ha comentado, LDA solo permite la extracción de $r - 1$ atributos, lo cual puede ser subóptimo en algunos casos, pues se podría perder bastante información.

### 6.1.3. ANMM

ANMM (*Average Neighbor Margin Maximization*) es una técnica de aprendizaje de métricas de distancia orientada específicamente a la reducción de dimensionalidad. Sigue por tanto el mismo camino que los ya comentados PCA y LDA, intentando solventar algunas de las limitaciones que presentan estos últimos.

El objetivo de ANMM es aprender una transformación lineal $L \in \mathcal{M}_{d' \times d}(\mathbb{R})$, con $d' \leq d$, que proyecte los datos a un espacio de menor dimensión, de forma que se maximice la similitud entre elementos de la misma clase y la separación entre elementos de distintas clases, siguiendo el criterio de maximización de márgenes que vamos a mostrar a continuación.

Consideramos el conjunto de datos de entrenamiento $\mathcal{X} = \{x_1, \ldots, x_N\} \subset \mathbb{R}^d$, con etiquetas $y_1, \ldots, y_N$, y fijamos $\xi, \zeta \in \mathbb{N}$, y la distancia euclídea como distancia inicial. A partir de estas variables vamos a construir dos tipos de vecindarios.

**Definición 6.1.1.** Sea $x_i \in \mathcal{X}$.

Se define el *$\xi$-vecindario homogéneo más cercano* de $x_i$ como el conjunto de los $\xi$ datos más cercanos a $x_i$ que están en su misma clase. Lo notaremos por $\mathcal{N}_i^o$.

Se define el *$\zeta$-vecindario heterogéneo más cercano* de $x_i$ como el conjunto de los $\zeta$ datos más cercanos a $x_i$ que están en clases distintas a la de $x_i$. Lo notaremos por $\mathcal{N}_i^e$.

Lo que va a tratar de maximizar ANMM es el concepto de margen promedio de vecindario, que definimos a continuación.

**Definición 6.1.2.** Dado $x_i \in \mathcal{X}$, se define su margen promedio de vecindario, y se nota $\gamma_i$, como

$$\gamma_i = \sum_{k\colon x_k \in \mathcal{N}_i^e} \frac{\|x_i - x_k\|^2}{|\mathcal{N}_i^e|} - \sum_{j\colon x_j \in \mathcal{N}_i^o} \frac{\|x_i - x_j\|^2}{|\mathcal{N}_i^o|}. \tag{6.7}$$

Se define el margen promedio (global) de vecindario como

$$\gamma = \sum_{i=1}^{N} \gamma_i. \tag{6.8}$$

Observemos que, para cada $x_i \in \mathcal{X}$, su margen promedio representa la diferencia entre la distancia media de $x_i$ a sus vecinos heterogéneos y la distancia media de $x_i$ a sus vecinos homogéneos. Por tanto, la maximización de este margen permite, localmente, alejar los datos de distintas clases y atraer a aquellos de la misma clase. En la figura 6.3 se describe gráficamente el concepto de margen promedio de vecindario.

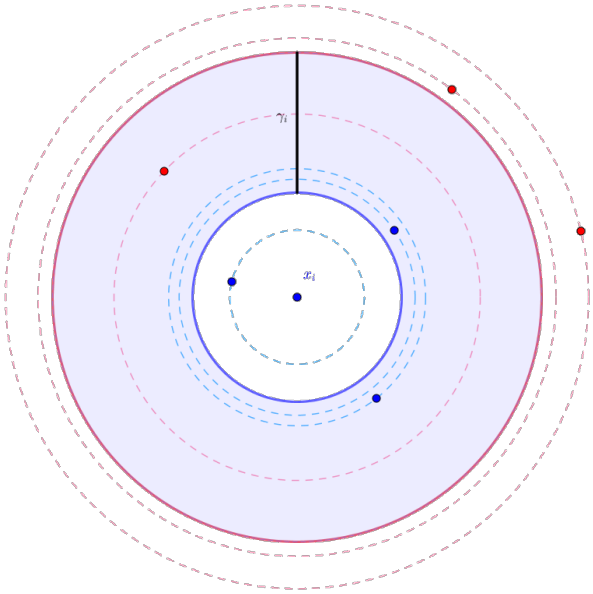


Figura 6.3: Descripción gráfica del margen promedio de vecindario para el dato $x_i$, para $\xi = \zeta = 3$. Las circunferencias azul y roja determinan la distancia media de $x_i$ a los datos de igual y distinta clase, respectivamente.

Buscamos ahora una transformación $L$ que maximice el margen asociado a los datos proyectados $\{Lx_i \colon i = 1, \dots, N\}$. Para tales datos, tenemos el margen asociado a dicha transformación,

$$\gamma^L = \sum_{i=1}^{N} \gamma_i^L = \sum_{i=1}^{N} \left( \sum_{k\colon x_k \in \mathcal{N}_i^e} \frac{\|Lx_i - Lx_k\|^2}{|\mathcal{N}_i^e|} - \sum_{j\colon x_j \in \mathcal{N}_i^o} \frac{\|Lx_i - Lx_j\|^2}{|\mathcal{N}_i^o|} \right).$$

Notemos que, gracias a la linealidad de la traza, podemos expresar

$$\begin{aligned} \sum_{i=1}^{n} \sum_{k\colon x_k \in \mathcal{N}_i^e} \frac{\|Lx_i - Lx_k\|^2}{|\mathcal{N}_i^e|} &= \operatorname{tr}\left( \sum_{i=1}^{N} \sum_{k\colon x_k \in \mathcal{N}_i^e} \frac{(Lx_i - Lx_k)(Lx_i - Lx_k)^T}{|N_i^e|} \right) \\ &= \operatorname{tr}\left[ L \left( \sum_{i=1}^{N} \sum_{k\colon x_k \in \mathcal{N}_i^e} \frac{(x_i - x_k)(x_i - x_k)^T}{|N_i^e|} \right) L^T \right] \\ &= \operatorname{tr}(LSL^T), \end{aligned}$$

donde $S = \sum_i \sum_{k\colon x_k \in \mathcal{N}_i^e} \frac{(x_i - x_k)(x_i - x_k)^T}{|\mathcal{N}_i^e|}$ recibe el nombre de matriz de dispersión. De la misma forma, si llamamos $C = \sum_i \sum_{j\colon x_j \in \mathcal{N}_i^o} \frac{(x_i - x_j)(x_i - x_j)^T}{|\mathcal{N}_i^o|}$, la cual denominaremos matriz de compacidad, se tiene que

$$\sum_{i=1}^{n} \sum_{j\,:\, x_j \in \mathcal{N}_i^o} \frac{\|Lx_i - Lx_j\|^2}{|\mathcal{N}_i^o|} = \mathrm{tr}(LCL^T).$$

Y por tanto, combinando ambas expresiones,

$$\gamma^L = \mathrm{tr}(L(S-C)L^T). \tag{6.9}$$

La maximización de $\gamma^L$ tal como se presenta en la fórmula 6.9 no es lo suficientemente restrictiva, pues basta multiplicar $L$ por constantes positivas para obtener un valor de $\gamma^L$ tan grande como queremos. Por eso, se añade la restricción $LL^T = I$, por lo que acabamos obteniendo el siguiente problema de optimización:

$$\begin{aligned} \max_{L \in \mathcal{M}_{d' \times d}(\mathbb{R})} \quad & \mathrm{tr}\left(L(S-C)L^T\right) \\ \text{s.a.:} \quad & LL^T = I \end{aligned}$$

Notemos que $S - C$ es simétrica, pues es la diferencia de dos matrices semidefinidas positivas (cada una de ellas es suma de productos tensoriales). El teorema 2.4.6 nos dice que la matriz $L$ que buscamos la podemos construir añadiendo, por filas, los $d'$ vectores propios correspondientes a los $d'$ mayores valores propios de $S - C$.

Notemos que ANMM solventa alguna de las carencias de los ya vistos PCA y LDA. Por un lado, se trata de un algoritmo de aprendizaje supervisado, luego utiliza la información de las clases que es ignorada por PCA. Y frente a las carencias de LDA, podemos observar que:

- No tiene problemas de cómputo con muestras pequeñas, para las cuales las matrices de dispersión o compacidad podrían resultar singulares, pues no tiene que calcular sus matrices inversas.
- No asume ninguna distribución sobre las clases.
- Admite cualquier tamaño para la reducción de dimensionalidad, no impone que dicho tamaño sea inferior al número de clases.

Por último, podemos observar también que, si mantenemos la dimensión máxima $d$, la condición $LL^T = I$ implica que $L$ es ortogonal y $L^T L = I$, luego estamos aprendiendo únicamente una isometría, como ya ocurría con PCA. Por ello, clasificadores basados en distancias como el kNN solo podrán experimentar mejoras cuando la dimensión escogida sea estrictamente menor que la original.

## 6.2. Técnicas orientadas a la mejora del clasificador de vecinos cercanos

### 6.2.1. LMNN

LMNN (*Large Margin Nearest Neighbors*) [37] es un algoritmo de aprendizaje de métricas de distancia orientado específicamente a mejorar la precisión del clasificador kNN. Se basa en la premisa de que el kNN clasificará con más fiabilidad un ejemplo si sus $k$ vecinos comparten la misma etiqueta, y para ello

intenta aprender una distancia que maximice el número de ejemplos que comparten etiqueta con el mayor número de vecinos posible.

De esta forma, el algoritmo LMNN trata de minimizar una función de error que penaliza, por un lado, las distancias grandes entre cada ejemplo y los considerados como sus vecinos ideales, y por otro lado, las distancias pequeñas entre ejemplos de distintas clases.

Supongamos que tenemos un conjunto de datos $\mathcal{X} = \{x_1, \dots, x_N\} \subset \mathbb{R}^d$ con etiquetas $y_1, \dots, y_N$. Para su funcionamiento, el algoritmo hace uso del concepto de *vecinos objetivo* o *target neighbors*. Dado un ejemplo $x_i \in \mathcal{X}$, sus $k$ vecinos objetivos son aquellos ejemplos de la misma clase que $x_i$, y distintos de este, para los que se desea que sean considerados como vecinos en la clasificación del kNN. Si $x_j$ es un vecino objetivo de $x_i$, entonces lo notaremos $j \rightsquigarrow i$. Estos vecinos objetivo están fijos durante el proceso de aprendizaje. Si se dispone de alguna información a priori se puede utilizar para determinarlos. En caso contrario, una buena opción es utilizar los vecinos cercanos de la misma clase para la distancia euclídea.

Una vez establecidos los vecinos objetivo, para cada distancia y para cada ejemplo que manejemos podemos establecer un perímetro determinado por el vecino más lejano a dicho ejemplo. Buscamos distancias para las cuales no haya ejemplos de otras clases en dicho perímetro. Hay que destacar que con este perímetro no hay suficientes garantías de separación, pues la distancia encontrada podría haber colapsado todos los vecinos objetivo en un punto y entonces el perímetro tendría radio cero. Por ello se considera un margen determinado por el radio del perímetro, al que se añade una constante positiva. Veremos que no hay pérdida de generalidad, por la función objetivo que vamos a definir, en suponer que dicha constante es 1. A cualquier ejemplo de distinta clase que invada este margen lo llamaremos *impostor*. Nuestro objetivo, por tanto, será, además de acercar cada ejemplo a sus vecinos objetivo lo máximo posible, intentar alejar lo máximo posible a los impostores.

En términos matemáticos, si nuestra distancia está determinada por la aplicación lineal $L \in \mathcal{M}_d(\mathbb{R})$, y $x_i, x_j \in \mathcal{X}$ con $j \rightsquigarrow i$, diremos que $x_l \in \mathcal{X}$ es un impostor para los datos anteriores si $y_l \neq y_i$ y $\|L(x_i - x_j)\|^2 \leq \|L(x_i - x_j)\|^2 + 1$. En la figura 6.4 se describen gráficamente los conceptos de vecino objetivo e impostor. Notemos por último que el margen está definido en términos de la distancia al cuadrado, en lugar de considerar solo la distancia. Esto facilitará la resolución del problema que vamos a formular.

A continuación, procedemos a definir de forma precisa los términos de la función objetivo. Como ya se ha mencionado, va a estar compuesta de dos términos. El primero penalizará a los vecinos objetivo lejanos y el segundo penalizará a los impostores cercanos. El primer término se define como

$$\varepsilon_{pull}(L) = \sum_{i=1}^{N} \sum_{j \rightsquigarrow i} \|L(x_i - x_j)\|^2.$$

Notemos que su minimización genera una fuerza de atracción entre los datos. El segundo término se define como

$$\varepsilon_{push}(L) = \sum_{i=1}^{N} \sum_{j \rightsquigarrow i} \sum_{l=1}^{N} (1 - y_{il})[1 + \|L(x_i - x_j)\|^2 - \|L(x_i - x_l)\|^2]_+,$$

donde $y_{il}$ es una variable binaria, que vale 1 y $y_i = y_l$ y 0 si $y_i \neq y_l$, y el operador $[\cdot]_+ \colon \mathbb{R} \to \mathbb{R}_0^+$ se define como $[z]_+ = \max\{z, 0\}$. De esta forma, este error suma cuando $y_{il} = 0$ (es decir, $x_l$ es de distinta clase que $x_i$), y el segundo factor es estrictamente positivo (es decir, se sobrepasa el margen definido

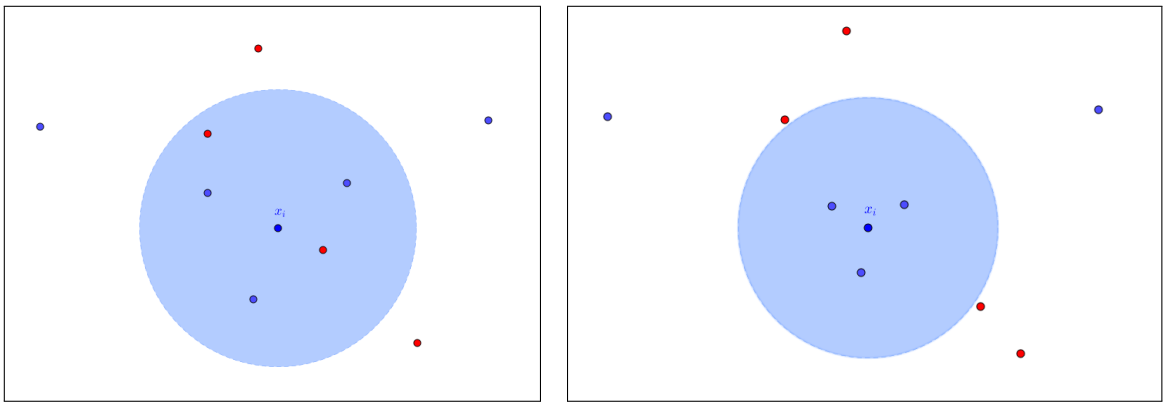


Figura 6.4: Descripción gráfica de vecinos objetivo e impostores (para $k = 3$) para el dato $x_i$. El círculo azul representa el margen que determinan los vecinos objetivo. Todos los puntos de distintas clases en dicho círculo son impostores. El objetivo LMNN será acercar los vecinos objetivos lo máximo posible y eliminar los impostores del círculo. Por tanto, no influirán los datos de la misma clase que no sean vecinos objetivo y se dejarán de penalizar los impostores en cuanto salgan del margen, como se muestra a la derecha. Esto da un carácter local a esta técnica de aprendizaje.

para los impostores). La minimización de este segundo término genera una fuerza de repulsión entre los datos.

Finalmente, la función objetivo resulta de combinar estos dos términos. Fijado $\mu \in ]0, 1[$, definimos

$$\varepsilon(L) = (1-\mu)\varepsilon_{pull}(L) + \mu\varepsilon_{push}(L). \tag{6.10}$$

Los autores afirman que, experimentalmente, la elección de $\mu$ no provoca grandes diferencias en los resultados, por lo que se suele tomar $\mu = 1/2$. La minimización de esta función nos llevará a aprender la distancia que buscábamos. Notemos que esta función no es convexa, por lo que si utilizamos un método de descenso bajo esta aproximación podemos quedar atrapados en un óptimo local. Sin embargo, podemos reformular nuestra función objetivo para que actúe sobre el cono de las matrices semidefinidas positivas. Si para cada $L \in \mathcal{M}_d(\mathbb{R})$, tomamos $M = L^T L \in \mathcal{M}_d(\mathbb{R})_0^+$, sabemos que $\|x_i - x_j\|_M^2 = \|x_i - x_j\|_L^2$, y por tanto,

$$\varepsilon(M) = (1-\mu)\sum_{i=1}^{N}\sum_{j\rightsquigarrow i}\|x_i - x_j\|_M^2 + \mu\sum_{i=1}^{N}\sum_{j\rightsquigarrow i}\sum_{l=1}^{N}[1 + \|x_i - x_j\|_M^2 - \|x_i - x_l\|_M^2]_+ \tag{6.11}$$

es una función convexa en $M$ que tiene toma los mismos valores que $\varepsilon(L)$. La minimización de $\varepsilon(M)$ en este caso está sujeta a la restricción $M \succeq 0$, por lo que podemos efectuarla mediante programación semidefinida, desplazándonos en la dirección del gradiente y proyectando el resultado sobre el cono semidefinido en sucesivas iteraciones. Además, podemos calcular un subgradiente $G \in \partial\varepsilon/\partial M$ dado por

$$G = (1-\mu)\sum_{i,j\rightsquigarrow i} O_{ij} + \mu\sum_{(i,j,l)\in\mathcal{N}}(O_{ij} - O_{il}),$$

donde $\mathcal{N}$ es el conjunto de tripletas $(i, j, l)$ para las cuales $x_l$ es un impostor sobre $x_i$ con el margen determinado por $x_j$, y $O_{ij} = (x_i - x_j)(x_i - x_j)^T$ son los productos tensoriales obtenidos de derivar las distancias. El primer término del gradiente es constante, mientras que el segundo solo varía en cada iteración con los cambios de los impostores que entran o salen de el conjunto $\mathcal{N}$. Estas consideraciones permiten realizar un cálculo del gradiente bastante eficiente.

En cuanto a la reducción de dimensionalidad, se presentan dos alternativas diferentes. Si mantenemos la optimización respecto a $M$, no es factible añadir restricciones de rango y seguir obteniendo un problema de programación semidefinida. Por tanto, se propone el uso de PCA previamente a la ejecución del algoritmo, para proyectar los datos sobre sus componentes principales, y aplicar LMNN sobre los datos proyectados. La otra alternativa es optimizar la función objetivo respecto a $L \in \mathcal{M}_{d' \times d(\mathbb{R})}$, con $d' < d$, usando algún algoritmo de gradiente descendente. En este caso la optimización no es convexa, pero aprendemos directamente una transformación lineal que reduce la dimensionalidad sin realizar cambios en el problema de optimización. Los autores afirman además, basados en los resultados empíricos, que esta optimización no convexa da buenos resultados.

Otras propuestas realizadas para la mejora de este algoritmo consisten en aplicar LMNN múltiples veces, aprendiendo así nuevas métricas cada vez, e ir utilizando estas métricas para determinar vecinos objetivo cada vez más precisos, o bien aprender métricas localmente. Por último, aunque la distancia aprendida está diseñada para que pueda ser utilizada por el kNN, también es posible utilizar la propia función objetivo como método para clasificar. Estos modelos de clasificación se denominan basados en energía. De este modo, para clasificar un dato test $x_t$, para cada posible valor de clase $y_t$, buscamos $k$ vecinos objetivo en el conjunto de entrenamiento de clase $y_t$, y evaluamos la *energía* para la métrica aprendida, asignando finalmente a $x_t$ el valor de $y_t$ que proporcione menor energía. De acuerdo con la funión objetivo, la energía penalizará distancias grandes entre $x_t$ y sus vecinos objetivo, los impostores en el perímetro de $x_t$ y perímetros de otras clases invadidas por $x_t$. Por tanto,

$$\begin{aligned} y_t^{pred} = \arg\min_{y_t} \Bigg\{ & (1-\mu) \sum_{j \rightsquigarrow t} \|x_t - x_j\|_M^2 \\ & + \mu \sum_{j \rightsquigarrow t, l} (1 - y_{tl}) \left[1 + \|x_t - x_j\|_M^2 - \|x_t - x_l\|_M^2\right]_+ \\ & + \mu \sum_{i, j \rightsquigarrow i} (1 - y_{it}) \left[1 + \|x_i - x_j\|_M^2 - \|x_i - x_t\|_M^2\right]_+ \Bigg\}. \end{aligned} \tag{6.12}$$

### 6.2.2. NCA

#### El kNN y la validación *Leave One Out*

En la mayoría de problemas de clasificación, para medir la eficacia del clasificador con el que estamos trabajando, se suele dividir el conjunto de datos del que disponemos en dos grupos: un conjunto de entrenamiento, que será el que utilice el clasificador para aprender, y un conjunto de validación, sobre el que el clasificador asigna sus predicciones, las cuales se comparan con las clases reales para evaluar el acierto del clasificador.

En algunos casos también puede resultar de interés evaluar el rendimiento del clasificador sobre los propios datos de entrenamiento, por ejemplo, para determinar si se está produciendo sobreaprendizaje, es decir, si el clasificador se adapta demasiado a los datos de entrenamiento, perdiendo así capacidad de generalización. Otra razón para ello es poder utilizar el rendimiento sobre los datos de entrenamiento como función objetivo a optimizar durante el proceso de aprendizaje. Esto contribuirá a mejorar el rendimiento del clasificador, siempre que no caiga en el sobreaprendizaje. Vamos a centrarnos en esta última razón.

En el caso del kNN, si pretendemos medir el rendimiento sobre los datos de entrenamiento, nos

encontramos con un inconveniente que nos puede llevar a una interpretación incorrecta de los resultados. Y es que, para cada dato en el conjunto de entrenamiento, su vecino más cercano es él mismo, y por tanto, la clase que vaya a serle asignada va a estar condicionada por este hecho. Esto se aprecia más claramente en el caso $k = 1$, donde el único vecino cercano considerado coincide siempre con el propio dato, y por tanto la tasa de acierto va a ser del 100 %.

La forma de solucionar este inconveniente consiste en, si $\mathcal{X}$ es el conjunto de datos de entrenamiento, para cada $x \in \mathcal{X}$, obtener su predicción encontrando sus $k$ vecinos más cercanos en $X \setminus \{x\}$. Esto es equivalente a particionar $X$ en conjuntos de un elemento, usando uno de los subconjuntos para la validación, y el resto para el entrenamiento. Este procedimiento de validación se conoce como validación cruzada *Leave One Out* (LOO). Como procedimiento de validación en general, su estimación del error es poco sesgada y no tiene componente aleatoria, aunque está sometido a mayor variabilidad y es más costoso computacionalmente que otras técnicas de validación.

### El análisis de componentes de vecindarios

NCA (*Neighborhood Component Analysis*) [15] es un algoritmo de aprendizaje de métricas de distancia orientado específicamente a mejorar la precisión del clasificador kNN. Tiene como finalidad aprender una transformación lineal cuyo objetivo principal es minimizar el error *Leave One Out* esperado por la clasificación mediante kNN. Adicionalmente, esta transformación podría usarse para reducir la dimensionalidad del conjunto de datos, y hacer por tanto más eficiente el clasificador.

Consideramos $\mathcal{X} = \{x_1, \dots, x_N\} \subset \mathbb{R}^d$ un conjunto de datos de entrenamiento con etiquetas $y_1, \dots, y_N$, respectivamente. Queremos aprender una distancia, determinada por una transformación lineal $L \in \mathcal{M}_d(\mathbb{R})$, que optimice la precisión del clasificador de vecinos cercanos. Lo ideal sería optimizar la actuación sobre los datos de validación, pero solo disponemos del conjunto de entrenamiento. Por tanto, nuestro objetivo va a ser optimizar el error *Leave One Out* de clasificación sobre el conjunto de entrenamiento.

Sin embargo, la función que para cada $L$ asigna el error LOO para la distancia asociada a $L$ no tiene garantías de diferenciabilidad, ni siquiera de continuidad, por lo que no es fácil tratar con ella para su optimización (notemos que esta función toma un conjunto finito de valores y está definida en un conjunto conexo, luego no puede ser continua a menos que sea constante, lo cual no sucede en ejemplos no triviales).

Para ello, NCA trata de abordar el problema de forma estocástica, esto es, en vez de operar con el error LOO directamente, lo hace sobre su valor esperado para la probabilidad que vamos a definir a continuación.

Dados dos ejemplos $x_i, x_j \in \mathcal{X}$, definimos la probabilidad de que $x_i$ tenga a $x_j$ como su vecino más cercano para la distancia $L$ como

$$p_{ij}^L = \frac{\exp\left(-\|Lx_i - Lx_j\|^2\right)}{\sum\limits_{k \neq i} \exp\left(-\|Lx_i - Lx_k\|^2\right)} \quad (j \neq i), \qquad p_{ii}^L = 0. \tag{6.13}$$

Notemos que, efectivamente, $p_{i*}$ define una medida de probabilidad sobre el conjunto $\{1, \dots, N\}$, para cada $i \in \{1, \dots, N\}$. Bajo esta ley de probabilidad, podemos definir la probabilidad de que el ejemplo $x_i$ esté correctamente clasificado como la suma de las probabilidades de que $x_i$ tenga como vecino más

cercano a cada ejemplo de su misma clase, esto es,

$$p_i^L = \sum_{j \in C_i} p_{ij}^L, \text{ donde } C_i = \{j \in \{1, \dots, N\} \colon y_j = y_i\}. \tag{6.14}$$

Finalmente, el número esperado de ejemplos correctamente clasificados, y la función que vamos a maximizar, la obtenemos como

$$f(L) = \sum_{i=1}^{N} p_i^L = \sum_{i=1}^{N} \sum_{j \in C_i} p_{ij}^L = \sum_{i=1}^{N} \sum_{\substack{j \in C_i \\ j \neq i}} \frac{\exp\left(-\|Lx_i - Lx_j\|^2\right)}{\sum\limits_{k \neq i} \exp\left(-\|Lx_i - Lx_k\|^2\right)}. \tag{6.15}$$

Esta función sí es diferenciable, y su derivada es

$$\frac{\partial f}{\partial L}(L) = 2L \sum_{i=1}^{N} \left( p_i^L \sum_{k=1}^{N} p_{ik}^L x_{ik} x_{ik}^T - \sum_{j \in C_i} p_{ij}^L x_{ij} x_{ij}^T \right). \tag{6.16}$$

Una vez obtenido el gradiente, podemos optimizar la función objetivo aplicando algún método de gradiente ascendente. Notemos que la función objetivo no es cóncava, y por tanto puede quedar atrapada en óptimos locales. Por otra parte, respecto al posible sobreajuste, los autores afirman que, basados en los resultados experimentales, no se produce sobreaprendizaje aunque se ascienda mucho en la función objetivo.

Finalmente, notemos que el mismo procedimiento es aplicable a cualquier matriz $L \in \mathcal{M}_{d' \times d}(\mathbb{R})$, con $d' < d$, por lo que NCA también puede ser utilizado para reducir la dimensionalidad de nuestro conjunto de datos.

## 6.3. Técnicas orientadas a la mejora del clasificador de centroides cercanos

### 6.3.1. NCMML

NCMML (*Nearest Class Mean Metric Learning*) [23] es un algoritmo de aprendizaje de métricas de distancia orientado a mejorar específicamente el clasificador NCM. Para ello, utiliza un enfoque probabilístico simiar al utilizado por NCA para mejorar la precisión del kNN.

Consideramos el conjunto de datos de entrenamiento $\mathcal{X} = \{x_1, \dots, x_N\} \subset \mathbb{R}^d$, con etiquetas $y_1, \dots, y_N \in \mathcal{C}$, donde $\mathcal{C} = \{c_1, \dots, c_r\}$ es el conjunto de clases. Para cada $c \in \mathcal{C}$, llamamos $\mu_c \in \mathbb{R}^d$ al vector media de los datos pertenecientes a la clase $c$, es decir, $\mu_c = \frac{1}{N_c} \sum_{i \colon y_i = c} x_i$, donde $N_c$ es el número de elementos de $\mathcal{X}$ que pertenecen a la clase $c$. Dada una transformación lineal $L \in \mathcal{M}_{d' \times d}(\mathbb{R})$, vamos a definir, para cada $x \in \mathcal{X}$ y cada $c \in \mathcal{C}$, la probabilidad de que $x$ sea etiquetado don la clase $c$ (de acuerdo con el criterio NCM) como

$$p_L(c|x) = \frac{\exp\left(-\frac{1}{2}\|L(x - \mu_c)\|^2\right)}{\sum\limits_{c' \in \mathcal{C}} \exp\left(-\frac{1}{2}\|L(x - \mu_{c'})\|^2\right)}. \tag{6.17}$$

Notemos que efectivamente $p_L(\cdot|x)$ define una probabilidad en el conjunto $\mathcal{C}$. Una vez definida la probabilidad anterior, la función objetivo que trata de maximizar NCMML es el logaritmo de la

verosimilitud para los datos etiquetados del conjunto de entrenamiento, esto es,

$$\mathcal{L}(L) = \frac{1}{N}\sum_{i=1}^{N} \log p_L(y_i|x_i). \tag{6.18}$$

Esta función es diferenciable y su gradiente viene dado por

$$\frac{\partial \mathcal{L}}{\partial L}(L) = \frac{1}{N}\sum_{i=1}^{N}\sum_{c\in\mathcal{C}} \alpha_{ic} L(\mu_c - x_i)(\mu_c - x_i)^T, \tag{6.19}$$

donde $\alpha_{ic} = p_L(c|x_i) - [\![y_i = c]\!]$ y $[\![\cdot]\!]$ denota la función indicadora de la condición $\cdot$. La maximización por métodos de gradiente de esta función es la tarea llevada a cabo por NCMML.

### 6.3.2. NCMC

#### Generalizando NCM: El clasificador de múltiples centroides

Aunque el clasificador NCM es un clasificador sencillo, intuitivo y eficiente tanto en el proceso de aprendizaje como el proceso de predicción, tiene un gran inconveniente, y es que presupone que las clases están agrupadas alrededor de su centro, lo cual es una hipótesis demasiado restrictiva. En la figura 6.5 podemos ver un ejemplo en el que NCM es incapaz de dar buenos resultados.

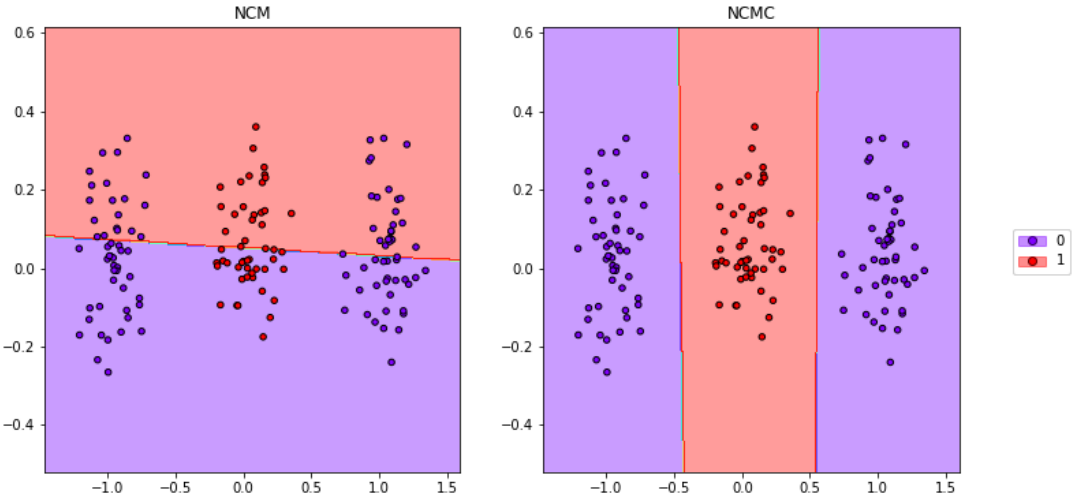


Figura 6.5: Conjunto de datos donde el clasificador NCM no da buenos resultados, pues los centroides de ambas clases son muy cercanos y ambos caen entre los puntos de la clase 1. Veremos que, escogiendo más de un centroide de forma adecuada podremos clasificar este conjunto como se muestra en la figura de la derecha.

Una forma de solventar este problema es, en vez de considerar el centro de la clase para clasificar nuevos datos, encontrar subgrupos dentro de cada clase que presenten un agrupamiento de calidad, y para cada uno de estos subgrupos considerar su centro. Tendríamos de esta forma un conjunto de centroides

para cada clase, y a la hora de clasificar un nuevo dato, bastaría seleccionar el centroide más cercano y asignarle la clase de la que es centroide.

Este nuevo clasificador, que denominaremos NCMC (*Nearest Class Multiple Centroids*), entran en juego los algoritmos de segmentación o *clustering*. Existen numerosos algoritmos [39] para obtener un conjunto de clusters dado un conjunto de datos, cada uno con sus ventajas e inconvenientes. Dada la forma de nuestro problema, en el que nos interesa además no solo obtener un conjunto de clusters para cada clase, sino además un centro para cada cluster, el algoritmo que reúne las condiciones más idóneas, además de ser sencillo y eficiente, es K-Means.

### K-Means y la búsqueda de centroides

*K-means* es uno de los algoritmos de *clustering* más populares. La idea original de este algoritmo es, fijado un natural $k$, encontrar $k$ clusters, cada uno con un centroide, que minimicen una función de coste que depende de las distancias de los puntos del cluster a su centroide. Encontrar la solución a ese problema de optimización es un problema NP-hard, incluso para aproximarlo. En consecuencia, se utiliza comúnmente un algoritmo iterativo que garantiza la reducción de la función de coste en cada iteración, si bien los resultados que ofrece no son necesariamente los óptimos. Es por ello que normalmente se denomina K-means a este algoritmo iterativo, en lugar de a la minimización de la función objetivo asociada.

En primer lugar describimos la función objetivo. Consideramos el conjunto de datos $\mathcal{X} = \{x_1, \dots, x_N\} \subset \mathbb{R}^d$. La función objetivo dependerá de una familia de subconjuntos, $C_1, \dots, C_k$ que forman una partición de $\mathcal{X}$. A su vez, cada conjunto $C_i$ tendrá asociado un centroide $\mu_i$, que será aquel punto del espacio euclídeo $\mathbb{R}^d$ que minimice la suma de los cuadrados de las distancias de los elementos en $C_i$ a dicho centroide. La función objetivo sumará los cuadrados de estas distancias para todos los clusters, quedando como se muestra a continuación:

$$G(C_1, \dots, C_k) = \min_{\mu_1, \dots, \mu_k \in \mathbb{R}^d} \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2. \tag{6.20}$$

A continuación se describe el algoritmo iterativo que se utiliza normalmente para la búsqueda de los centroides. Inicialmente, se parte de unos centroides $\mu_1^{(0)}, \dots, \mu_k^{(0)}$ escogidos al azar. En cada iteración $t$, se determina la partición $\{C_1^{(t)}, \dots, C_k^{(t)}$ a partir de los centroides $\mu_1^{(t-1)}, \dots, \mu_k^{(t-1)}$ de forma que a cada $x \in \mathcal{X}$ se le asigna el cluster del centroide más cercano. Finalmente, se calculan los nuevos centroides $\mu_1^{(t)}, \dots, \mu_k^{(t)}$ como el vector media de los datos en $\{C_1^{(t)}, \dots, C_k^{(t)}$, respectivamente. El proceso se repite hasta que en una iteración no se produzca ningún cambio en los clusters generados, momento en el que el algoritmo habrá convergido.

Como ya se ha mencionado, este algoritmo garantiza que la función objetivo 6.20 decrece conforme aumenta el número de iteraciones. A pesar de esto, no se puede asegurar que el algoritmo alcance un óptimo global (en ocasiones es posible incluso que ni siquiera se alcance un óptimo local). Sin embargo, el carácter aleatorio y la eficiencia del algoritmo permite realizar distintas ejecuciones eligiendo diferentes centroides iniciales, lo que facilita el encuentro de soluciones aceptables.

Aunque se trata de un algoritmo eficiente y que en la práctica obtiene buenos resultados en la función objetivo, hay que destacar que es necesario especificar el número de clusters previamente. Los clusters que se obtienen con este algoritmo suelen tomar formas esféricas o similares, no siendo útiles para datos

que presentan otros tipos de agrupaciones. También los datos más lejanos pueden condicionar en gran medida el valor de los centroides, y en consecuencia también el de los clusters.

Para la clasificación con NCMC, el uso de K-Means se reduce a aplicar el algoritmo de segmentación dentro de cada subconjunto de datos asociado a cada una de las clases del problema de clasificación. De esta forma obtenemos de forma sencilla el conjunto de centroides buscado para cada clase, y sobre el cual podemos realizar la clasificación de nuevos datos buscando simplemente el centroide más cercano. De nuevo se hace necesario establecer previamente el número de centroides para cada clase. Dichos números pueden estimarse realizando validación cruzada.

#### Aprendiendo distancias para NCMC

Una vez definido el clasificador NCMC, el proceso de aprendizaje de distancias es análogo al de NCM. Siguiendo la notación utilizada en NCMML, en este caso, en lugar de un conjunto de centros de clase $\{\mu_c\}$, con $c \in \mathcal{C}$, disponemos de un conjunto de centroides, $\{m_{c_j}\}_{j=1}^{k_c}$, con $k_c \in \mathbb{N}$, para cada $c \in \mathcal{C}$. En este caso, las probabilidades asociadas a cada clase para la predicción correcta de $x \in \mathcal{X}$ vienen dadas por $p_L(c|x) = \sum_{j=1}^{k_c} p_L(m_{c_j}|x)$, donde son los centroides aquellos cuya probabilidad viene dada por la función softmax

$$p_L(m_{c_j}|x) = \frac{\exp\left(-\frac{1}{2}\|L(x - m_{c_j})\|^2\right)}{\sum\limits_{c\in\mathcal{C}}\sum\limits_{i=1}^{k_c}\exp\left(-\frac{1}{2}\|L(x - m_{c_i})\|^2\right)}. \tag{6.21}$$

De nuevo, maximizamos el logaritmo de la verosimilitud, $\mathcal{L}(L) = \frac{1}{N}\sum_{i=1}^{N} p_L(y_i|x_i)$, cuyo gradiente en este caso viene dado por

$$\frac{\partial\mathcal{L}}{\partial L}(L) = \frac{1}{N}\sum_{i=1}^{N}\sum_{c\in\mathcal{C}}\sum_{j=1}^{k_c}\alpha_{ic_j}L(m_{c_j} - x_i)(m_{c_j} - x_i)^T,$$

donde $\alpha_{ic_j} = p_L(m_{c_j}|x_i) - [\![y_i = c]\!]\frac{p_L(m_{c_j}|x_i)}{\sum_{j'=1}^{k_c} p_L(m_{c_{j'}}|x_i)}$. La maximización de la verosimilitud por métodos de gradiente es la tarea llevada a cabo por la técnica de aprendizaje de distancias para NCMC, que denominaremos con el mismo nombre que dicho clasificador.

## 6.4. Técnicas basadas en teoría de la información

### 6.4.1. ITML

ITML (*Information Theoretic Metric Learning*) [12] es una técnica de aprendizaje de métricas de distancia cuyo objetivo es encontrar una métrica lo más cercana posible a una distancia de partida, entendiendo la cercanía desde el punto de vista de la entropía relativa, como formularemos más adelante, haciendo que dicha métrica satisfaga determinadas restricciones de similitud para los datos entrenados.

ITML parte de un conjunto de datos $\mathcal{X} = \{x_1, \ldots, x_N\} \subset \mathbb{R}^d$, no necesariamente etiquetados, pero del que se conoce que determinados pares de datos considerados similares deben estar a una distancia menor o igual que $u$, y otros pares de datos considerados no similares deben estar situados a una distancia

mayor o igual que $l$, donde $u$ y $l$ son constantes prefijadas de antemano, con valores relativamente pequeño y grande, respectivamente, respecto al conjunto de datos.

A partir de los datos con las restricciones indicadas, ITML considera una métrica inicial asociada a una matriz $M_0$ definida positiva, y trata de encontrar una matriz definida positiva $M$, lo más parecida posible a $M_0$, y que respete las restricciones de similitud impuestas. La forma de medir el parecido entre $M$ y $M_0$ se realiza utilizando herramientas de la teoría de la información.

Es conocido que hay una correspondencia entre las matrices definidas positivas y las distribuciones normales multivariante, fijado un mismo vector media $\mu$. Dada $M \in \mathcal{M}_d(\mathbb{R})^+$, podemos construir una distribución normal a través de su función de densidad,

$$p(x; M) = \frac{1}{(2\pi)^{n/2} \det(M)^{1/2}} \exp\left((x-\mu)^T M^{-1}(x-\mu)\right).$$

Recíprocamente, a partir de dicha distribución, si calculamos la matriz de covarianza recuperamos la matriz $M$. Usando esta correspondencia, vamos a medir la cercanía entre $M_0$ y $M$ a través de la divergencia KL entre sus correspondientes gaussianas,

$$\mathrm{KL}(p(x; M_0))\|p(x; M)) = \int p(x; M_0) \log \frac{p(x; M_0)}{p(x; M)} dx.$$

Una vez definido el mecanismo con el que vamos a medir la cercanía de las métricas, podemos establecer la formulación del problema a optimizar por la técnica ITML. Si denominamos $S$ y $D$ a los conjuntos de pares de índices sobre los elementos de $\mathcal{X}$ que representan a los datos considerados similares y no similares, respectivamente, y partimos de la métrica inicial $M_0$, el problema es

$$\begin{aligned} \min_{M \succeq 0} \quad & \mathrm{KL}(p(x; M_0)\|p(x; M)) \\ \text{s.a.:} \quad & d_M(x_i, x_j) \leq u, \quad (i,j) \in S \\ & d_M(x_i, x_j) \geq l, \quad (i,j) \in D. \end{aligned} \tag{6.22}$$

Para tratar este problema computacionalmente, se hace uso de la divergencia matricial *log-det*, la cual viene dada por

$$D_{ld}(M\|M_0) = \mathrm{tr}(MM_0^{-1}) - \log\det(MM_0^{-1}) - d, \quad M, M_0 \in \mathcal{M}_d(\mathbb{R})^+.$$

Hemos visto en el teorema 3.3.3 que la divergencia KL entre dos gaussianas con la misma media se puede expresar en términos de la divergencia *log-det* como

$$\mathrm{KL}(p(x; M_0)\|p(x; M)) = \frac{1}{2} D_{ld}(M_0\|M).$$

Esto nos permite reformular el problema 6.22 de la siguiente forma:

$$\begin{aligned} \min_{M \succeq 0} \quad & D_{ld}(M_0\|M) \\ \text{s.a.:} \quad & \mathrm{tr}(M(x_i - x_j)(x_i - x_j)^T) \leq u, \quad (i,j) \in S \\ & \mathrm{tr}(M(x_i - x_j)(x_i - x_j)^T) \geq l, \quad (i,j) \in D. \end{aligned} \tag{6.23}$$

Es posible que no se pueda encontrar una métrica que satisfaga simultáneamente todas las restricciones, por lo que el problema podría no tener solución. Por ello, ITML introduce en el problema 6.23 variables

de holgura mediante las cuales se obtiene un problema en cuya optimización se establece un equilibrio entre la minimización de la divergencia y la satisfacción de las restricciones, para poder llegar así a una solución aproximada del problema original, en caso de no tener solución. Finalmente, la técnica computacional utilizada en la resolución de este problema de optimización es la conocida como el método de las *proyecciones de Bregman* [4], el cual es una generalización del método de las proyecciones iteradas.

### 6.4.2. DMLMJ

DMLMJ (*Distance Metric Learning through the maximization of the Jeffrey divergence*) [25] es una técnica de aprendizaje de métricas de distancia basada, al igual que ITML, en conceptos de teoría de la información. Concretamente, la herramienta que utiliza es la conocida como divergencia de Jeffrey, a través de la cual pretende separar lo máximo posible la distribución asociada a los puntos similares de aquella asociada a los puntos no similares, en los términos que veremos a continuación.

Consideramos el conjunto de entrenamiento $\mathcal{X} = \{x_1, \dots, x_N\} \subset \mathbb{R}^d$ con correspondientes etiquetas $y_1, \dots, y_N$, y fijamos $k \in \mathbb{N}, k \geq 1$. Como ya hemos comentado, DMLMJ busca maximizar, respecto a la divergencia de Jeffrey, la separación entre las distribuciones de puntos similares y no similares. Para ello, introduciremos los siguientes conceptos:

**Definición 6.4.1.** Dado $x_i \in \mathcal{X}$, el *vecindario k-positivo* de $x_i$ se define como el conjunto de los $k$ vecinos más cercanos de $x_i$ en $\mathcal{X} \setminus \{x_i\}$ cuya clase es la misma que la de $x_i$. Se nota por $V_k^+(x_i)$.

El *vecindario k-negativo* de $x_i$ se define como el conjunto de los $k$ vecinos más cercanos de $x_i$ en $\mathcal{X}$ cuya clase es distinta de la de $x_i$. Se nota por $V_k^-(x_i)$.

Se define el *espacio de diferencias k-positivo* del conjunto de datos etiquetados como el conjunto

$$S = \{x_i - x_j \colon x_i \in \mathcal{X}, x_j \in V_k^+(x_i)\}.$$

Análogamente, se define el *espacio de diferencias k-negativo* como el conjunto

$$D = \{x_i - x_j \colon x_i \in \mathcal{X}, x_j \in V_k^-(x_i)\}.$$

Los conjuntos $S$ y $D$ representan, por tanto, los vectores con las diferencias entre los datos y sus $k$ vecinos más cercanos, de igual o de distinta clase, respectivamente. Llamamos $P$ y $Q$ a las distribuciones en los espacios $S$ y $D$, respectivamente, asumiendo que son gaussianas multivariante. Asumiremos, además, que ambas distribuciones tienen media 0. Esta asunción es razonable, puesto que en la práctica, en la mayoría de los casos, si $x_i$ es vecino de $x_j$, $x_j$ también lo es de $x_i$, luego ambas diferencias aparecerán en el espacio de diferencias, haciendo la media cero. Por último, llamaremos a las correspondientes matrices de covarianza $\Sigma_S$ y $\Sigma_D$, respectivamente.

Si ahora aplicamos una transformación lineal a los datos, $x \mapsto Lx$, con $L \in \mathcal{M}_{d' \times d}(\mathbb{R})$, las distribuciones transformadas seguirán teniendo media 0 y covarianzas $L\Sigma_S L^T$ y $L\Sigma_D L^T$, respectivamente. A dichas distribuciones las denominaremos $P_L$ y $Q_L$. El objetivo de DMLMJ es encontrar una transformación que maximice la divergencia de Jeffrey entre $P_L$ y $Q_L$:

$$\max_{L \in \mathcal{M}_{d' \times d}(\mathbb{R})} \quad f(L) = \mathrm{JF}(P_L \| Q_L) = \mathrm{KL}(P_L \| Q_L) + \mathrm{KL}(Q_L \| P_L).$$

Como se demostró en la proposición 3.3.5, la divergencia de Jeffrey entre las distribuciones gaussianas $P_L$ y $Q_L$ se puede expresar como

$$f(L) = \frac{1}{2}\operatorname{tr}\left((L\Sigma_S L^T)^{-1}(L\Sigma_D L^T) + (L\Sigma_D L^T)^{-1}(L\Sigma_S L^T)\right) - d'.$$

Como $d'$ es constante, obtenemos el problema equivalente

$$\max_{L\in\mathcal{M}_{d'\times d}(\mathbb{R})} \quad J(L) = \operatorname{tr}\left((L\Sigma_S L^T)^{-1}(L\Sigma_D L^T) + (L\Sigma_D L^T)^{-1}(L\Sigma_S L^T)\right).$$

Si calculamos el gradiente de la función anterior, obtenemos

$$\nabla J(L) = \left(2\Sigma_D L^T\Sigma_{2S}^{-1} - 2\Sigma_S L^T\Sigma_{2S}^{-1}\Sigma_{2D}\Sigma_{2S}^{-1}\right) + \left(2\Sigma_S L^T\Sigma_{2D}^{-1} - 2\Sigma_D L^T\Sigma_{2D}^{-1}\Sigma_{2S}\Sigma_{2D}^{-1}\right),$$

donde $\Sigma_{2S} = L\Sigma_S L^T$ y $\Sigma_{2D} = L\Sigma_D L^T$.

Cada uno de los términos entre paréntesis se anula, respectivamente, cuando $\Sigma_S^{-1}\Sigma_D L^T = L^T\Sigma_{2S}^{-1}\Sigma_{2D}$ y cuando $\Sigma_D^{-1}\Sigma_S L^T = L^T\Sigma_{2D}^{-1}\Sigma_{2S}$. Queremos resolver estas dos ecuaciones. Para ello, enunciamos los siguientes resultados.

**Teorema 6.4.1.** *Sean $\Sigma_1, \Sigma_2 \in \mathcal{M}_d(\mathbb{R})^+$ y sea $L \in \mathcal{M}_{d'\times d}(\mathbb{R})$ una matriz que contiene, por filas, $d'$ vectores propios linealmente independientes de $\Sigma_1^{-1}\Sigma_2$. Entonces,*

$$\Sigma_1^{-1}\Sigma_2 L^T = L^T(L\Sigma_1 L^T)^{-1}(L\Sigma_2 L^T). \tag{6.24}$$

*Demostración.* Sea $D \in \mathcal{M}_{d'}(\mathbb{R})$ la matriz diagonal que contiene, en el mismo orden, los $d'$ valores propios de $\Sigma_1^{-1}\Sigma_2$ asociados a los vectores propios presentes en $L$. Se verifica entonces

$$\Sigma_1^{-1}\Sigma_2 L^T = L^T D. \tag{6.25}$$

Si multplicamos a ambos lados por $L\Sigma_1$, obtenemos

$$L\Sigma_2 L^T = L\Sigma_1 L^T D.$$

$L\Sigma_1 L^T$ es una matriz regular. Puesto que $\Sigma_1$ es definida positiva, admite una descomposición $\Sigma_1 = PP^T$, con $P$ regular de dimensión $d$. Entonces, $L\Sigma_1 L^T = LPP^T L^T = LP(LP)^T$, donde $LP$ es una matriz cuadrada de dimensión $d'$ y de rango $d'$, por ser la composición (viéndolas como aplicaciones lineales) de un isomorfismo con una aplicación de rango $d'$, haciendo que la imagen tenga de nuevo dimensión $d'$. En consecuencia, $LP$ es regular, y por tanto también lo es $LP(LP)^T = L\Sigma_1 L^T$, por lo que podemos tomar inversas, obteniendo

$$(L\Sigma_1 L^T)^{-1}(L\Sigma_2 L^T) = D. \tag{6.26}$$

Si a continuación multiplicamos a la izquierda por $L^T$ en ambos lados de la igualdad, se tiene

$$L^T(L\Sigma_1 L^T)^{-1}(L\Sigma_2 L^T) = L^T D.$$

Sustituyendo lo que acabamos de obtener en el lado derecho de 6.25, concluimos que

$$\Sigma_1^{-1}\Sigma_2 L^T = L^T(L\Sigma_1 L^T)^{-1}(L\Sigma_2 L^T).$$

□

**Corolario 6.4.2.** *Sean $\Sigma_1, \Sigma_2 \in \mathcal{M}_d(\mathbb{R})^+$ y sea $L \in \mathcal{M}_{d' \times d}(\mathbb{R})$ una matriz que contiene, por filas, $d'$ vectores propios linealmente independientes de $\Sigma_1^{-1}\Sigma_2$. Sea $D \in \mathcal{M}_{d'}(\mathbb{R})$ la matriz diagonal con los correspondientes $d'$ valores propios. Entonces,*

$$\operatorname{tr}\left((L\Sigma_1 L^T)^{-1}(L\Sigma_2 L^T)\right) = \operatorname{tr}(D).$$

*Demostración.* Basta tomar trazas en la igualdad de matrices de la ecuación 6.26. □

De acuerdo con el teorema 6.4.2, podemos observar que es posible anular simultáneamente ambos términos del gradiente tomando $L$ como una matriz de vectores propios de $\Sigma_S^{-1}\Sigma_D$, o bien de $\Sigma_D^{-1}\Sigma_S$. Notemos que ambos casos comparten los mismos vectores propios y cada opción tiene como valores propios los inversos de la opción restante. Optamos por tomar $d'$ vectores propios del primer caso, de $\Sigma_S^{-1}\Sigma_D$. Entonces, $\nabla J(L) = 0$ y el corolario 6.4.2 nos dice que

$$\begin{aligned} J(L) &= \operatorname{tr}\left((L\Sigma_S L^T)^{-1}(L\Sigma_D L^T) + (L\Sigma_D L^T)^{-1}(L\Sigma_S L^T)\right) \\ &= \operatorname{tr}\left((L\Sigma_S L^T)^{-1}(L\Sigma_D L^T)\right) + \operatorname{tr}\left((L\Sigma_D L^T)^{-1}(L\Sigma_S L^T)\right) \\ &= \operatorname{tr}(\Lambda) + \operatorname{tr}\left(\Lambda^{-1}\right) = \sum_{i=1}^{d'} \left(\lambda_i + \frac{1}{\lambda_i}\right), \end{aligned}$$

donde $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_{d'})$ es la matriz diagonal con los valores propios asociados a los vectores propios escogidos de $\Sigma_S^{-1}\Sigma_D$.

Por tanto, para maximizar $J(L)$ debemos seleccionar los $d'$ vectores propios asociados a los $d'$ valores propios $\lambda$ para los cuales sea mayor la expresión $\lambda + 1/\lambda$. La transformación $L$ construida a partir de dichos vectores propios determina la distancia que es aprendida mediante la técnica DMLMJ.

Finalmente, el único requerimiento adicional necesario para completar esta construcción es el cálculo de las matrices de covarianza. Teniendo en cuenta que se ha asumido que la media de las distribuciones es 0, podemos obtener dichas matrices de forma sencilla a partir de los vectores de diferencias, como se muestra a continuación.

$$\Sigma_S = \frac{1}{|S|}\sum_{i=1}^{N}\left[\sum_{x_j \in V_k^+(x_i)} (x_i - x_j)(x_i - x_j)^T\right], \quad \Sigma_D = \frac{1}{|D|}\sum_{i=1}^{N}\left[\sum_{x_j \in V_k^-(x_i)} (x_i - x_j)(x_i - x_j)^T\right].$$

### 6.4.3. MCML

MCML (*Maximally Collapsing Metric Learning*) [14] es una técnica de aprendizaje de métricas de distancia supervisado, que se basa en la idea de que si todos los datos de una misma clase fueran proyectados a un mismo punto, y los datos de distintas clases fueran proyectados en puntos distintos y suficientemente alejados, tendríamos, sobre los datos proyectados, una separación de clases ideal. Su finalidad es aprender una métrica de distancia que permita colapsar lo máximo posible, dentro de las limitaciones de la métrica, todos los datos de una misma clase en un único punto, arbitrariamente alejado de los puntos en los que colapsarán los datos del resto de las clases.

Consideramos el conjunto de datos $\mathcal{X} = \{x_1, \ldots, x_N\} \subset \mathbb{R}^d$, con etiquetas asociadas $y_1, \ldots, y_N$. Queremos aprender una métrica determinada por una matriz $M \succeq 0$ que trate de colapsar al máximo las clases según el enfoque del párrafo anterior. La forma de abordar este problema va a consistir una vez más en utilizar las herramientas proporcionadas por la teoría de la información. Para ello, en primer

lugar, introducimos una distribución condicionada sobre los puntos del conjunto de datos en forma de *softmax*, análoga a la establecida en el caso de NCA. Si $i, j \in \{1, \dots, N\}$ con $i \neq j$, definimos la probabilidad de que $x_j$ sea clasificado con la clase de $x_i$ según la distancia entre $x_i$ y $x_j$ que determina $M$ como

$$p^M(j|i) = \frac{\exp(-\|x_i - x_j\|_M^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|_M^2)}. \tag{6.27}$$

Por otra parte, la distribución ideal que buscamos obtener es una distribución binaria para la que la probabilidad de que a un dato se le asigne su clase correcta es 1, y 0 en caso contrario, es decir,

$$p_0(j|i) \propto \begin{cases} 1, & y_i = y_j \\ 0, & y_i \neq y_j \end{cases}. \tag{6.28}$$

Notemos que durante el entrenamiento conocemos las clases reales de los datos, luego podemos tratar esta última probabilidad. Además, observemos que si conseguimos una métrica $M$ cuyas distribuciones $p^M$ asociadas coincidan con $p_0$, entonces, bajo unas mínimas condiciones de suficiencia de datos, estaremos consiguiendo colapsar las clases en puntos infinitamente alejados.

En efecto, supongamos que hay al menos $r + 2$ datos en cada clase, donde $r$ es el rango de $M$, y que $p^M(j|i) = p_0(j|i)$ para cualesquiera $i, j \in \{1, \dots, N\}$. Entonces, por un lado, de $p^M(j|i) = 0$ para $y_i \neq y_j$ se deduce que $\exp(-\|x_i - x_j\|_M^2) = 0$, lo que conduce indudablemente a que $x_i$ y $x_j$ estén infinitamente alejados cuando sus clases son distintas. Por otra parte, de $p^M(j|i) \propto 1$ para cualesquiera $x_i, x_j$ con $y_i = y_j$ se deduce que el valor $\exp(\|x_i - x_j\|_M^2)$ es constante para todos los miembros de una misma clase, y en consecuencia todos los puntos en una misma clase son equidistantes. Como $M$ tiene rango $r$ está induciendo una distancia sobre un subespacio de dimensión $r$, donde se sabe que a lo sumo puede haber $r + 1$ puntos distintos y equidistantes entre sí (esto se debe a que los puntos distintos equidistantes para una distancia procedente de un producto escalar han de ser afinmente independientes). Como estamos asumiendo que hay al menos $r + 2$ puntos por clase, todos los puntos de la misma clase han de tener distancia 0 entre sí respecto a $M$, colapsando por tanto en un único punto.

Una vez establecidas ambas distribuciones, el objetivo de MCML es, como ya hemos comentado, aproximar $p^M(\cdot|i)$ a $p_0(\cdot|i)$ lo máximo posible, para cada $i$ utilizando para ello la entropía relativa entre ambas distribuciones. El problema de optimización consiste, por tanto, en minimizar esta divergencia,

$$\min_{M \succeq 0} \quad f(M) = \sum_{i=1}^{N} \mathrm{KL}\left[p_0(j|i) \| p^M(j|i)\right]. \tag{6.29}$$

Podemos reescribir la función objetivo en términos de funciones más elementales.

$$\begin{aligned} f(M) &= \sum_{i=1}^{N} \sum_{j=1}^{N} p_0(j|i) \log \frac{p_0(j|i)}{p^M(j|i)} = \sum_{i=1}^{N} \sum_{j\colon y_i = y_j} \log \frac{1}{p^M(j|i)} \\ &= \sum_{i=1}^{N} \sum_{j\colon y_i = y_j} -\log p^M(j|i) = -\sum_{i=1}^{N} \sum_{j\colon y_i = y_j} \left( -\|x_i - x_j\|_M^2 - \log \sum_{k \neq i} \exp(-\|x_i - x_k\|^2) \right) \\ &= \sum_{i=1}^{N} \sum_{j\colon y_i = y_j} \|x_i - x_j\|_M^2 + \sum_{i=1}^{N} \log \sum_{k \neq i} \exp(-\|x_i - x_k\|^2). \end{aligned} \tag{6.30}$$

Cada sumando de la expresión anterior es convexo en $M$, el primero por tratarse de una función distancia en $M$ (que es lineal), y el segundo por tratarse de una función logaritmo de suma de exponenciales (convexo) compuesto con una función distancia. Además, la restricción $M \succeq 0$ también es convexa, luego el problema a optimizar es convexo. La función objetivo es siempre no negativa, pues $\|x_i - x_j\|_M^2 \geq -\log \sum_{k \neq i} \exp(-\|x_i - x_k\|^2)$, pues el lado derecho de la desigualdad siempre contiene al menos una exponencial con el término $-\|x_i - x_j\|_M^2$ y el logaritmo es creciente. Luego el problema es convexo y minorado, lo que nos garantiza que se alcanza un mínimo global y no hay mínimos locales. Podemos por tanto encontrar dicho óptimo utilizando los mecanismos de la optimización convexa.

Concretamente, la técnica de optimización propuesta en MCML es la ya comentada programación semidefinida. Para ello, es necesario una expresión del gradiente de la función objetivo, el cual puede ser calculado a partir de su expresión en 6.30:

$$\nabla f(M) = \sum_{i,j\colon y_i = y_j} (x_i - x_j)^T (x_i - x_j) - \sum_i \frac{-\sum\limits_{k \neq i} (x_i - x_k)^T (x_i - x_k) \exp(-\|x_i - x_k\|_M^2)}{\sum\limits_{k \neq i} \exp(-\|x_i - x_k\|_M^2)}.$$

La minimización de la función objetivo 6.30 de forma iterativa mediante descensos en la dirección del gradiente combinado con proyecciones sobre el cono de las matrices semidefinidas positivas es la tarea llevada a cabo por el algoritmo MCML.

## 6.5. Otras técnicas de aprendizaje de métricas de distancia

### 6.5.1. LSI

LSI (*Learning with Side Information*) [38], también denominado a veces MMC (*Mahalanobis Metric for Clustering*) es una técnica de aprendizaje de métricas de distancia que trabaja con un conjunto de datos, no necesariamente etiquetados, sobre los que se conoce, como información adicional, determinadas parejas de puntos que se sabe que son similares y, opcionalmente, parejas de puntos que se sabe que no lo son.

LSI trata de aprender una métrica $M$ que respete esta información adicional. Es por ello que puede ser utilizado tanto en aprendizaje supervisado, donde los pares similares corresponderán a datos con la misma etiqueta, como en aprendizaje no supervisado con restricciones de similaridad, como por ejemplo, problemas de clustering donde se conoce que determinados datos deben ser agrupados en el mismo cluster.

Pasamos a formular el problema a optimizar. Supongamos que nuestro conjunto de datos es $\mathcal{X} = \{x_1, \ldots, x_N\} \subset \mathbb{R}^d$ y conocemos adicionalmente el conjunto $S = \{(x_i, x_j) \in \mathcal{X} \times \mathcal{X} \colon x_i \text{ y } x_j \text{ son similares.}\}$. De forma adicional, podemos conocer el conjunto $D = \{(x_i, x_j) \in \mathcal{X} \times \mathcal{X} \colon x_i \text{ y } x_j \text{ no son similares.}\}$. En caso de no disponer de este último, podemos tomar $D$ como el complemento de $S$ en $\mathcal{X} \times \mathcal{X}$.

La primera intuición para abordar este problema, dada la información de la que disponemos, es minimizar las distancias entre los pares de puntos similares, esto es, minimizar $\sum_{(x_i,x_j) \in S} \|x_i - x_j\|_M^2$, donde $(x_i - x_j)^T M (x_i - x_j)$ y $M \in \mathcal{M}_d(\mathbb{R})_0^+$. Sin embargo, esto nos conduce a la solución $M = 0$, lo cual no nos aportaría ninguna información productiva. Por eso, LSI añade la restricción adicional

$\sum_{(x_i,x_j)\in D} \|x_i - x_j\|_M \geq 1$, lo que nos conduce al problema de optimización

$$\begin{aligned} \min_{M} \quad & \sum_{(x_i,x_j)\in S} \|x_i - x_j\|_M^2 \\ \text{s.a.:} \quad & \sum_{(x_i,x_j)\in D} \|x_i - x_j\|_M \geq 1 \\ & M \succeq 0. \end{aligned} \tag{6.31}$$

Notemos varias observaciones respecto a esta fórmula. En primer lugar, la elección de la constante 1 en la restricción es irrelevante; si escogemos cualquier constante $c > 0$ obtenemos una métrica proporcional a $M$. Por otra parte, el problema de optimización es convexo, pues los conjuntos determinados por las restricciones son convexos y la función a optimizar también lo es. Por último, podríamos plantearnos una restricción sobre el conjunto $D$ de la forma $\sum_{(x_i,x_j)\in D} \|x_i - x_j\|_M^2 \geq 1$. Sin embargo, un razonamiento similar al utilizado en LDA permite probar que en tal caso la métrica aprendida va a tener rango 1, lo cual puede no resultar óptimo.

Para la resolución de este problema los autores proponen el problema equivalente

$$\begin{aligned} \max_{M} \quad & \sum_{(x_i,x_j)\in D} \|x_i - x_j\|_M \\ \text{s.a.:} \quad & \sum_{(x_i,x_j)\in S} \|x_i - x_j\|_M^2 \leq 1 \\ & M \succeq 0. \end{aligned} \tag{6.32}$$

Este problema con dos restricciones convexas puede ser resuelto mediante métodos de gradiente ascendente combinados con proyecciones que satisfagan las restricciones del problema. En este problema, las restricciones son fáciles de satisfacer por separado. La primera restricción consiste en una proyección sobre un semiespacio afín, mientras que la segunda consiste en una proyección sobre el cono de matrices semidefinidas positivas. El método de las proyecciones iteradas justificado por el teorema 1.3.1 permite satisfacer ambas restricciones proyectando repetidas veces sobre ambos conjuntos, hasta obtener la convergencia.

### 6.5.2. DML-eig

DML (*Distance Metric Learning with Eigenvalue Optimization*) [40] es una técnica de aprendizaje de métricas de distancia inspirada en la técnica LSI de la sección anterior, proponiendo un problema de optimización muy similar pero ofreciendo un método de resolución completamente diferente, basado en la optimización de valores propios.

Consideramos, al igual que en el caso anterior, un conjunto de datos de entrenamiento $\mathcal{X} = \{x_1, \ldots, x_N\} \subset \mathbb{R}^d$, del que conocemos dos subconjuntos $S$ y $D$ de $\mathcal{X} \times \mathcal{X}$ de datos considerados similares, y no similares, respectivamente. En la sección anterior, para optimizar el problema 6.32 se proponía un método de gradiente ascendente con proyecciones iteradas, el cual puede requerir bastante tiempo para converger. La propuesta de DML-eig consiste en una ligera modificación de la función objetivo, manteniendo las

restricciones, que nos conduce al problema

$$\begin{aligned} \max_{M} \quad & \min_{(x_i,x_j)\in D} \|x_i - x_j\|_M^2 \\ \text{s.a.:} \quad & \sum_{(x_i,x_j)\in S} \|x_i - x_j\|_M^2 \leq 1 \\ & M \succeq 0. \end{aligned} \tag{6.33}$$

Para abordar la resolución de este problema resulta útil introducir una notación que simplifique la indexación de los datos. En primer lugar, denotaremos $X_{ij} = (x_i - x_j)(x_i - x_j)^T$ a los productos tensoriales entre las diferencias de elementos de $\mathcal{X}$. Para acceder a pares de elementos $(i, j)$ utilizaremos un único índice $\tau \equiv (i, j)$. Dicho índice lo podremos suponer ordenado cuando sea necesario para acceder a las distintas componentes de un vector de dimensión adecuada. Al producto tensorial anterior $X_{ij}$ también lo podremos notar como $X_\tau$. Por último, los conjuntos $S$ y $D$ también supondremos que están formados por índices $\tau$ asociados a un par $(i, j)$ tales que $x_i$ y $x_j$ son similares o no similares, respectivamente. De esta forma, si denotamos $X_S = \sum_{(i,j)\in S} X_{ij}$, el problema 6.33 podemos reescribirlo en términos del producto escalar de Frobenius como

$$\begin{aligned} \max_{M} \quad & \min_{\tau\in D} \langle X_\tau, M\rangle \\ \text{s.a.:} \quad & \langle X_S, M\rangle \leq 1 \\ & M \succeq 0. \end{aligned} \tag{6.34}$$

Veamos cómo se establece la formulación del problema que buscamos en términos de optimización de valores propios. Para cada matriz simétrica $X \in S_d(\mathbb{R})$ denotamos a su mayor valor propio como $\lambda_{\max}(X)$. Asociado al conjunto $D$ de pares no similares vamos a definir el simplex

$$\Delta = \left\{ u \in \mathbb{R}^{|D|} \colon u_\tau \geq 0 \ \forall \tau \in D, \sum_{\tau\in D} u_\tau = 1 \right\}.$$

También consideramos el conjunto

$$\mathcal{P} = \{M \in \mathcal{M}_d(\mathbb{R})_0^+ \colon \operatorname{tr}(M) = 1\}.$$

$\mathcal{P}$ es la intersección del cono de matrices semidefinidas positivas con un subespacio afín de $\mathcal{M}_d(\mathbb{R})$. A los conjuntos que presentan esta estructura se les conoce como *espectraedros*.

Entonces, si $X_S$ es definida positiva y definimos, para cada $\tau \in D$, $\widetilde{X}_\tau = X_S^{-1/2} X_\tau X_S^{-1/2}$, se puede probar [40] que el problema 6.34 es equivalente al siguiente problema

$$\max_{S\in\mathcal{P}} \min_{u\in\Delta} \sum_{\tau\in D} u_\tau \langle \widetilde{X}_\tau, S\rangle, \tag{6.35}$$

el cual se puede reescribir a su vez como un problema de optimización de valores propios:

$$\min_{u\in\Delta} \max_{S\in\mathcal{P}} \left\langle \sum_{\tau\in D} u_\tau \widetilde{X}_\tau, S \right\rangle = \min_{u\in\Delta} \lambda_{\max} \left( \sum_{\tau\in D} u_\tau \widetilde{X}_\tau \right). \tag{6.36}$$

El problema de minimizar el mayor valor propio de una matriz simétrica es conocido y se conocen algoritmos iterativos que permiten alcanzar dicho mínimo [26]. Además, en [40] se propone un algoritmo para resolver el problema $\max_{S\in\mathcal{P}} \min_{u\in\Delta} \sum_{\tau\in D} u_\tau \langle \widetilde{X}_\tau, S\rangle + \mu \sum_{\tau\in D} u_\tau \log u_\tau$, donde $\mu > 0$ es un parámetro de suavizado, mediante el cual se puede aproximar el problema 6.36.

### 6.5.3. LDML

LDML (*Logistic Discriminant Metric Learning*) [16] es una técnica de aprendizaje de métricas de distancia en cuyo modelo de optimización se hace uso de la función logística. Los autores afirman que esta técnica es bastante útil para aprender distancias sobre conjuntos de imágenes etiquetadas, pudiendo utilizarse por tanto en problemas como la identificación de caras.

Recordamos que la función sigmoide es la aplicación $\sigma\colon \mathbb{R} \to \mathbb{R}$ dada por

$$\sigma(x) = \frac{1}{1+e^{-x}}.$$

Esta función presenta una gráfica con una forma sigmoidal, es diferenciable, estrictamente creciente y toma valores entre 0 y 1, alcanzando dichos valores en sus límites en menos infinito y más infinito, respectivamente. Estas propiedades permiten que la función logística sea la función de distribución de una variable aleatoria, lo que le da una importante utilidad probabilística. Su gráfica presenta un comportamiento asintótico a partir de valores pequeños (en valor absoluto), con un crecimiento exponencial en zonas cercanas al cero. Esto hace que sea de gran utilidad para modelar señales binarias. También presenta una derivada fácil de calcular, y expresable en términos de la propia función logística, $\sigma'(x) = \sigma(x)(1-\sigma(x))$. En la figura 6.6 se muestra la gráfica de esta función. La función logística es también de gran utilidad en otras ramas del aprendizaje automático, como es el caso de la regresión logística o de las redes neuronales.

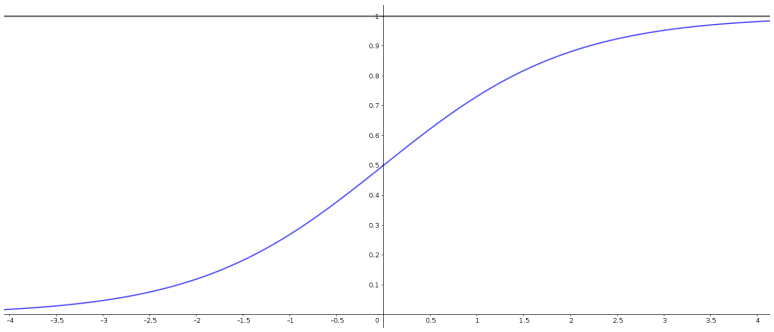

Figura 6.6: La función logística.

Supongamos el conjunto de datos $\mathcal{X} = \{x_1, \ldots, x_N\} \subset \mathbb{R}^d$, con correspondientes etiquetas $y_1, \ldots, y_N$. En LDML, la función logística se utiliza para definir una probabilidad, la cual asignará a pares de puntos una probabilidad mayor conforme menor sea la distancia entre ellos. Para medir la distancia, LDML utilizará una matriz de métrica $M$ semidefinida positiva, quedando la expresión de la probabilidad como

$$p_{ij,M} = \sigma(b - \|x_i - x_j\|_M^2), \tag{6.37}$$

donde $b$ es un valor umbral positivo que determinará el valor máximo alcanzable por la función logística, y que puede ser estimado mediante validación cruzada. Asociada a esta probabilidad podemos definir una variable aleatoria que sigue una distribución de Bernouilli, y que toma los valores 0 y 1, según el par $(x_i, x_j)$ pertenezca o no a la misma clase. Dicha distribución viene determinada por la función masa de probabilidad

$$f_{ij,M}(x) = (p_{ij,M})^x (1-p_{ij,M})^{1-x}, \quad x \in \{0,1\}.$$

La función que busca maximizar la técnica LDML es el logaritmo de la verosimilitud de la distribución anterior para el conjunto de datos dado, esto es,

$$\mathcal{L}(M) = \sum_{i,j=1}^{N} y_{ij} \log p_{ij,M} + (1 - y_{ij}) \log(1 - p_{ij,M}), \tag{6.38}$$

donde $y_{ij}$ es una variable binaria que toma el valor 1 si $y_i = y_j$, y 0 en caso contrario. Esta función es diferenciable, cóncava (es una combinación positiva de funciones que se pueden expresar como un menos logaritmo de suma de exponenciales, que son cóncavos) y está mayorada, luego tenemos la garantía de poder alcanzar un máximo global. Teniendo en cuenta que, si $x_{ij} \equiv (x_i - x_j)^T (x_i - x_j)$ y $p_{ij} \equiv p_{ij,M}$, y por las propiedades de la derivada de la función logística su gradiente presenta la expresión

$$\begin{aligned}
\nabla\mathcal{L}(M) &= \sum_{i,j=1}^{N} y_{ij} \frac{-x_{ij} p_{ij}(1 - p_{ij})}{p_{ij}} + (1 - y_{ij}) \frac{x_{ij} p_{ij}(1 - p_{ij})}{1 - p_{ij}} \\
&= \sum_{i,j=1}^{N} -y_{ij} x_{ij}(1 - p_{ij}) + (1 - y_{ij}) x_{ij} p_{ij} \\
&= \sum_{i,j=1}^{N} x_{ij}((1 - y_{ij}) p_{ij} - (1 - p_{ij}) y_{ij}) \\
&= \sum_{i,j=1}^{N} x_{ij}(p_{ij} - y_{ij}),
\end{aligned}$$

los métodos iterativos de gradiente ascendente combinados con proyecciones sobre el cono de las matrices semidefinidas positivas, conforman el algoritmo de programación semidefinida que es utilizado en LDML para la obtención de la métrica que optimiza su función objetivo.

## 6.6. El kernel trick. Algoritmos de aprendizaje de métricas de distancia basados en kernels

### 6.6.1. El kernel trick

Los métodos de kernel conforman un paradigma dentro del aprendizaje automático que resulta de gran utilidad en muchos de los problemas que se abordan en esta disciplina. Normalmente surgen en problemas en los que el algoritmo de aprendizaje ve mermada su capacidad, generalmente, debido a la forma del conjunto de datos. Esto ocurre, por ejemplo, en las máquinas de vectores soporte. Aunque no vamos a entrar en los detalles de este algoritmo, nos va a servir para ilustrar la necesidad de los métodos de kernel en el aprendizaje automático.

Las *máquinas de vectores soporte* (SVM, *Support Vector Machines*) son un modelo de clasificación lineal binario que, en su versión más sencilla, cuando los datos son separables, busca establecer el hiperplano que mejor que mejor separa los datos, esto es, aquel para el cual se maximiza la distancia (margen) a los conjuntos que determinan cada clase. Los puntos de ambos conjuntos donde se materializa dicho margen son los denominados *vectores soporte*. En la figura 6.7 se muestra cómo actúa este clasificador.

Sin embargo, en muchas ocasiones, aunque los datos sean visiblemente separables, puede resultar imposible separarlos mediante un hiperplano, como sucede en el ejemplo de la figura 6.8. En ella, se ha

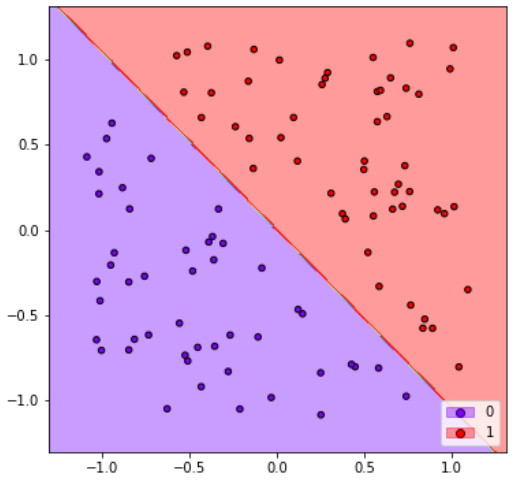


Figura 6.7: Clasificación realizada por las máquinas de vectores soporte en su versión básica.

considerado un subconjunto finito $\mathcal{X}$ de números reales, de forma que a aquellos $x \in \mathcal{X}$ con $|x| > 1$ se les ha asignado la clase 1 y a aquellos con $|x| \leq 1$ la clase $-1$. Es inmediato observar que, aunque las regiones de las dos clases están claramente diferenciadas, no es posible separarlas mediante un hiperplano.

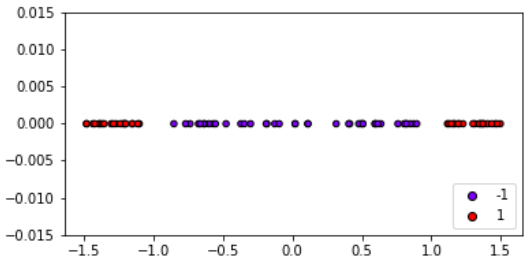


Figura 6.8: Conjunto de datos para el que SVM no puede establecer un hiperplano separador.

A pesar de esto, es posible establecer una transformación en los datos que los incluya en un espacio de dimensión mayor en el cual los datos sí sean separables mediante hiperplanos. En efecto, si definimos la aplicación $\phi\colon \mathbb{R} \to \mathbb{R}^2$ por $\phi(x) = (x, x^2)$, los datos de $\phi(\mathcal{X})$ sí que son separables en $\mathbb{R}^2$. Concretamente, podemos tomar el hiperplano $H = \{(x, y) \in \mathbb{R}^2\colon y = 1\}$ como hiperplano separador. Esta misma idea podemos utilizarla sobre SVM: transformamos los datos y aplicamos el algoritmo en el espacio transformado obteniendo el hiperplano. Si después queremos predecir la clase de un nuevo dato, podemos aplicarle la transformación y asignarle la clase según el lado del hiperplano en el que ha caído. La figura 6.9 muestra el resultado de aplicar SVM en el espacio transformado sobre el ejemplo dado.

En general, cuando nos encontramos con este tipo de problemas siempre podemos enviar los datos a un nuevo espacio de dimensión mayor, definiendo una aplicación $\phi\colon \mathcal{X} \to \mathcal{F}$. El espacio $\mathcal{F}$ lo tomamos como espacio de Hilbert y recibe el nombre de *espacio de características*. Aunque con el ejemplo

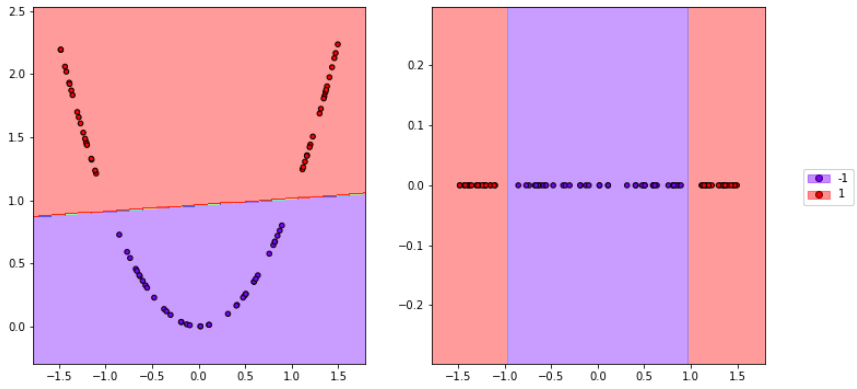


Figura 6.9: Resolución mediante máquinas de vectores soporte del problema de la figura 6.8 en el espacio de características. A la derecha se muestra el efecto del clasificador aprendido en el espacio de características sobre el conjunto de datos original.

anterior hemos podido comprobar el potencial de esta herramienta, tiene un gran inconveniente, y es que enviar los datos a un espacio de características puede aumentar en gran medida la dimensión del problema, y por lo tanto la aplicación de los algoritmos en el espacio de características puede ser muy costoso computacionalmente. Además, si quisiéramos trabajar en espacios de características de dimensión infinita sería imposible tratar los datos computacionalmente de esta forma. Para solventar estos problemas surge el concepto de *kernel trick*.

Si $\phi\colon \mathcal{X} \to \mathcal{F}$ es la aplicación de envío al espacio de características, se define la *función kernel* asociada como la forma bilineal simétrica $K\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ que determina los productos escalares de los datos en el espacio de características, esto es, $K(x, x') = \langle \phi(x), \phi(x') \rangle$. La clave del éxito de las funciones kernel es que, en muchos algoritmos, como ocurre en el caso de SVM, no es necesario tratar con los valores de los datos, sino únicamente con los productos escalares entre ellos. Por tanto, conociendo la función kernel disponemos de la información necesaria para trabajar en el espacio de características, sea cual sea su dimensión. Notemos por último que, si $X = \{x_1, \ldots, x_N\}$, el tamaño, la función kernel puede verse como una matriz $K \in S_N(\mathbb{R})$ donde $K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$, por lo que la complejidad del problema va a depender únicamente del tamaño del conjunto de datos, independientemente de la dimensión del espacio de características.

A continuación vamos a ver algunas de las funciones kernel más utilizadas. Con ellas podremos observar también cómo es posible utilizar las funciones kernel para trabajar en espacios de dimensión infinita.

Ejemplo 6.6.1 (El kernel lineal): El kernel lineal es el caso más sencillo de kernel, y viene representado por la función

$$K(x, x') = \langle x, x' \rangle.$$

Se corresponde con la transformación identidad sobre el mismo espacio de partida. △

Ejemplo 6.6.2 (Kernels polinómicos): Los kernels polinómicos de grado $k$ vienen dados por funciones kernel de la forma

$$K(x, x') = (\gamma \langle x, x' \rangle + c_0)^k.$$

Fijado $k \in \mathbb{N}$, veamos que en efecto $K$ es una función kernel, es decir, que hay una transformación $\phi$ para

la cual $K(x,x') = \langle \phi(x), \phi(x') \rangle$. Supongamos $\gamma = c_0 = 1$ (los cálculos son análogos para cualquier valor de estos parámetros). Además, supongamos $x = (x_1, \dots, x_d), x' = (x'_1, \dots, x'_d)$ y notamos $x_0 = x'_0 = 1$. Utilizamos tambiénla notación multiíndice de los polinomios en varias variables, de forma que si $\alpha = (\alpha_1, \dots, \alpha_d) \in (\mathbb{N} \cup \{0\})^d$, la expresión $x_\alpha$ representa el monomio $x_{\alpha_1} \dots x_{\alpha_d}$. Entonces se tiene

$$K(x,x') = \prod_{i=1}^{k} (1 + \langle x, x' \rangle) = \prod_{i=1}^{k} \sum_{j=0}^{d} x_j x'_j = \sum_{\alpha \in \{0,\dots,d\}^k} x_\alpha x'_\alpha$$

Si ahora definimos la aplicación $\phi \colon \mathbb{R}^d \to \mathbb{R}^{(d+1)^k}$ como

$$\phi(x) = (x_{(0,\dots,0)}, x_{(0,\dots,1)}, \dots, x_{(0,\dots,n)}, x_{(0,\dots,1,0)}, \dots, x_{(n,\dots,n)}),$$

se concluye que

$$\langle \phi(x), \phi(x') \rangle = \sum_{\alpha \in \{0,\dots,d\}^k} x_\alpha x'_\alpha = K(x,x').$$

Por tanto, $K$ es una función kernel y está construida a partir de transformaciones polinómicas de grado máximo $k$. △

Ejemplo 6.6.3 (El kernel gaussiano): El kernel gaussiano viene determinado por la aplicación $K \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ dada por

$$K(x,x') = \exp(-\gamma \|x - x'\|^2).$$

Es el kernel más popular junto al polinómico. También se le conoce como kernel RBF (*Radial Basis Funcion*), pues las funciones que definen los productos escalares dependen únicamente de la distancia entre los datos.

Veamos que, efectivamente, $K$ es una función kernel y que el espacio de características tiene dimensión infinita. Supongamos, por simplicidad, que el espacio de partida es $\mathbb{R}$. Entonces,

$$\begin{aligned} K(x,x') &= e^{-\gamma(x-x')^2} = e^{-\gamma x^2 + 2\gamma x x' - \gamma x'^2} \\ &= e^{-\gamma x^2 - \gamma x'^2} e^{2\gamma x x'} \\ &= e^{-\gamma x^2 - \gamma x'^2} \sum_{n=0}^{\infty} \frac{(2\gamma x x')^n}{n!} \\ &= e^{-\gamma x^2 - \gamma x'^2} \sum_{n=0}^{\infty} \sqrt{\frac{2\gamma}{n!}} x^n \sqrt{\frac{2\gamma}{n!}} x'^n \\ &= \langle \phi(x), \phi(x') \rangle_{l_2}, \end{aligned}$$

donde $l_2$ es el espacio de Hilbert las sucesiones de cuadrado sumables y $\phi \colon \mathbb{R} \to l_2$ es la aplicación dada por

$$\phi(x) = e^{-\gamma x^2} \left\{ \sqrt{\frac{2\gamma}{n!}} x^n \right\}_{n=0}^{\infty}.$$

Notemos que $\phi$ está bien definida, pues la sucesión que define tiene como suma de cuadrados una exponencial. Por tanto, con el kernel gaussiano obtenemos un espacio de características de dimensión infinita. △

Ejemplo 6.6.4 (Otros kernels):

a) Kernel laplaciano:
$$K(x, x') = \exp(-\gamma\|x - x'\|_1).$$

b) Kernel sigmoidal:
$$K(x, x') = \tanh(\gamma\langle x, x'\rangle + c_0).$$

c) Kernel cosenoidal:
$$K(x, x') = \frac{\langle x, x'\rangle}{\|x\|\|x'\|}.$$

En general, toda matriz semidefinida positiva es la matriz de una función kernel. Recíprocamente, las matrices de las funciones kernel son siempre semidefinidas positivas. △

En el aprendizaje de métricas de distancia, la utilidad de los kernels se debe a las limitaciones que vienen dadas por las distancias de Mahalanobis aprendidas. Aunque las métricas aprendidas pueden utilizarse posteriormente para aprender mediante clasificadores no lineales, como el kNN, las métricas en sí hemos visto que vienen determinadas por una aplicación lineal. Estas siempre vienen determinadas por la imagen de una base de vectores en el espacio de partida, lo que condiciona a que en la transformación resultante del aprendizaje se tenga la libertad únicamente de elegir las imágenes de tantos datos como dimensión tenga el espacio, transformando el resto de vectores por linealidad. Cuando la cantidad de datos es mucho mayor que la dimensión de su espacio esto puede convertirse en una limitación.

El uso de las funciones kernel es factible en muchos de los algoritmos del aprendizaje de métricas de distancia gracias a que, como ocurría con las máquinas de vectores soporte, muchos de estos algoritmos solo necesitan tratar con los productos escalares entre los datos, en lugar de hacerlo directamente con los datos. A modo de ejemplo, vamos a ver que es posible calcular distancias euclídeas, una herramienta esencial en muchos de los algoritmos, en el espacio de características, utilizando únicamente productos escalares entre los datos. En efecto, si $K$ es la matriz de la función kernel, $\phi\colon \mathcal{X} \to \mathcal{F}$ es la transformación al espacio de características y $x_i, x_j \in \mathcal{X}$, se tiene

$$\begin{aligned}\|\phi(x_i) - \phi(x_j)\|^2 &= \langle \phi(x_i) - \phi(x_j), \phi(x_i) - \phi(x_j)\rangle \\ &= \langle \phi(x_i), \phi(x_i)\rangle - 2\langle \phi(x_i), \phi(x_j)\rangle + \langle \phi(x_j), \phi(x_j)\rangle \\ &= K_{ii} + K_{jj} - 2K_{ij}.\end{aligned} \tag{6.39}$$

El siguiente problema común a todos los algoritmos de aprendizaje de métricas basados en kernels consiste en cómo tratar la transformación aprendida. En este caso, el aprendizaje consistirá en aprender una aplicación lineal en el espacio de características $L\colon \mathcal{F} \to \mathbb{R}^d$ que, como ocurre en los algoritmos ya estudiados, inducirá una distancia en el espacio de destino. El problema en esta situación es que $\mathcal{F}$ podría ser de muy alta dimensionalidad, o incluso de dimensión infinita, luego $L$ podría no ser tratable matricialmente. Retomando el ejemplo de las máquinas de vectores soporte, surge un problema similar, pues el hiperplano aprendido puede representarse mediante un vector $w$, que de nuevo podría tener dimensión muy alta o infinita. En este caso, se conocen *teoremas de representación* que permiten expresar $w$ como combinación lineal de los datos en el espacio de características, y dicha combinación lineal permite utilizar las funciones kernel para la transformación.

En el aprendizaje de métricas de distancia, si suponemos la aplicación lineal que queremos aprender $L$ también continua (es decir, $L \in \mathcal{L}(\mathcal{F}, \mathbb{R}^d)$), podemos expresarla como un vector de funcionales lineales y continuos, los cuales, por el teorema de representación de Riesz, están determinados por el producto escalar por un determinado vector, es decir, $L = (\langle \cdot, w_1\rangle, \ldots, \langle \cdot, w_d\rangle)$. Para los algoritmos que vamos a

estudiar, se conocen diversos teoremas de representación [5, 18, 24, 25, 30] que permiten expresar los vectores $w_i$ como combinación lineal de los datos en el espacio de características, esto es, para cada $i \in \{1, \dots, d'\}$ existe un vector $\alpha^i = (\alpha_1^i, \dots, \alpha_N^i) \in \mathbb{R}^N$ tal que $w_i = \sum_{j=1}^N \alpha_j^i \phi(x_j)$. En consecuencia, se verifica que

$$L\phi(x) = A \begin{pmatrix} K(x_1, x) \\ \vdots \\ K(x_N, x) \end{pmatrix},$$

donde $A \in \mathcal{M}_{d' \times N}(\mathbb{R})$ viene dada por $A_{ij} = \alpha_j^i$.

Gracias a estos teoremas, podemos tratar el problema computacionalmente si somos capaces de calcular los coeficientes de la matriz $A$. A la hora de transformar los datos de $\mathcal{X}$ bastará con multiplicar $A$ por la columna adecuada de la matriz del kernel. Si queremos transformar nuevos datos podemos de la misma forma definir una matriz (en este caso, no necesariamente cuadrada) con los productos escalares entre los datos de entrenamiento y los nuevos datos. De nuevo, escogiendo la columna adecuada en dicha matriz podremos transformar los datos según la aplicación definida por $L$.

Cada técnica de aprendizaje de métricas que admita el uso de kernels utilizará herramientas distintas para su funcionamiento, cada una de ellas basada en los algoritmos originales. En las siguientes secciones se describirán las kernelizaciones de algunos de los algoritmos estudiados.

### 6.6.2. KLMNN

KLMNN [34] es la versión kernelizada de LMNN. En ella, los datos del conjunto $\mathcal{X}$ se envían al espacio de características para aprender en dicho espacio una distancia que minimice la función de error establecida en el problema de LMNN.

Aunque el problema formulado en la versión no kernelizada se realizó respecto a una matriz de métrica $M$, mediante la función de error 6.11, a la hora de trabajar en espacios de características nos interesará más trabajar con una aplicación lineal, aunque se pierda la convexidad del problema, para poder utilizar el teorema de representación. Por tanto, adaptando al espacio de características la función de error propuesta en 6.10, el problema formulado para la versión kernelizada consiste en

$$\begin{aligned} \min_{L \in \mathcal{L}(\mathcal{F}, \mathbb{R}^d)} \quad & \varepsilon(L) = (1-\mu) \sum_{i=1}^N \sum_{j \rightsquigarrow i} \|L(\phi(x_i) - \phi(x_j))\|^2 \\ & + \mu \sum_{i=1}^N \sum_{j \rightsquigarrow i} \sum_{l=1}^N (1 - y_{il})[1 + \|L(\phi(x_i) - \phi(x_j))\|^2 - \|L(\phi(x_i) - \phi(x_l))\|^2]_+. \end{aligned} \tag{6.40}$$

Como consecuencia del teorema de representación, se verifica que, para cada $x_i \in \mathcal{X}$, $L\phi(x_i) = AK_{\cdot i}$, donde $A \in \mathcal{M}_{d' \times N}(\mathbb{R})$ es la matriz dada por el teorema de representación, y $K_{\cdot i}$ es la $i$-ésima columna de la matriz de kernel. Utilizando esto en la expresión del error, obtenemos

$$\begin{aligned} & (1-\mu) \sum_{i=1}^N \sum_{j \rightsquigarrow i} \|L(\phi(x_i) - \phi(x_j))\|^2 + \mu \sum_{i=1}^N \sum_{j \rightsquigarrow i} \sum_{l=1}^N (1 - y_{il})[1 + \|L(\phi(x_i) - \phi(x_j))\|^2 - \|L(\phi(x_i) - \phi(x_l))\|^2]_+ \\ & = (1-\mu) \sum_{i=1}^N \sum_{j \rightsquigarrow i} \|A(K_{\cdot i} - K_{\cdot j})\|^2 + \mu \sum_{i=1}^N \sum_{j \rightsquigarrow i} \sum_{l=1}^N (1 - y_{il})[1 + \|A(K_{\cdot i} - K_{\cdot j})\|^2 - \|A(K_{\cdot i} - K_{\cdot l})\|^2]_+. \end{aligned}$$

La expresión anterior depende únicamente de $A$ y de las funciones kernel, y minimizándola como función en $A$ (la notamos $\varepsilon(A)$) obtenemos el mismo valor que al minimizar $\varepsilon(L)$. Observemos también que la expresión $\varepsilon(A)$ requiere el cálculo de los vecinos objetivo y los impostores, pero estos dependen únicamente de las distancias en el espacio de características, las cuales ya hemos visto que son calculables, como se indica en la igualdad 6.39. Por tanto, todos los aspectos de $\varepsilon(A)$ son tratables computacionalmente, así que si aplicamos métodos de gradiente sobre $\varepsilon(A)$ podremos reducir el valor de la función objetivo, siempre teniendo en cuenta que podemos quedar atrapados en óptimos locales, pues el problema no es convexo. Finalmente, una vez encontrada una matriz $A$ que minimiza $\varepsilon(A)$, tendremos determinada la aplicación $L$ asociada gracias al teorema de representación, y podemos usar $A$ junto con las funciones kernel para transformar nuevos datos.

### 6.6.3. KANMM

KANMM [36] es la versión kernelizada de ANMM. En ella, los datos del conjunto $\mathcal{X}$ se envían al espacio de características mediante la aplicación $\phi\colon \mathcal{X} \to \mathcal{F}$. En dicho espacio aplicamos ANMM para obtener la aplicación lineal buscada.

Recordamos que el primer paso necesario para la aplicación de ANMM era la obtención de los vecindarios homogéneo y heterogéneo para cada dato $x_i \in \mathcal{X}$. Observemos que para este cálculo únicamente es necesario comparar distancias en el espacio de características, lo cual hemos visto que se puede realizar gracias a la función kernel, mediante la igualdad 6.39. Notaremos a los vecindarios en el espacio de características como $N^o_{\phi(x_i)}$ y $N^e_{\phi(x_i)}$, respectivamente, para cada $x_i$.

Las matrices (o endomorfismos, más en general) de dispersión y compacidad en el espacio de características vienen dados por

$$S^\phi = \sum_{i,k\colon \phi(x_k)\in N^e_{\phi(x_i)}} \frac{(\phi(x_i)-\phi(x_k))(\phi(x_i)-\phi(x_k))^T}{|N^e_{\phi(x_i)}|}$$
$$C^\phi = \sum_{i,j\colon \phi(x_j)\in N^o_{\phi(x_i)}} \frac{(\phi(x_i)-\phi(x_j))(\phi(x_i)-\phi(x_j))^T}{|N^o_{\phi(x_i)}|}.$$

El problema a optimizar se expresa, por tanto, como

$$\begin{aligned} \max_{L\in\mathcal{L}(\mathcal{F},\mathbb{R}^{d'})} \quad & \operatorname{tr}\left(L(S^\phi - C^\phi)L^T\right) \\ \text{s.a.:} \quad & LL^T = I \end{aligned} \tag{6.41}$$

De acuerdo con los teoremas de representación, cada uno de los vectores $w_i, i = 1,\dots,d'$ de $\mathcal{F}$ que caracterizan $L$ verifican $w_i = \sum_{j=1}^N \alpha_j^i \phi(x_j)$. En consecuencia, $L\phi(x_i) = AK_{\cdot i}$, donde $A$ es la matriz de coeficientes del teorema de representación y $K_{\cdot i}$ representa la $i$-ésima columna de la matriz del kernel. Entonces,

$$L(\phi(x_i)-\phi(x_j))(\phi(x_i)-\phi(x_j))^T L^T = A(K_{\cdot i}-K_{\cdot j})(K_{\cdot i}-K_{\cdot j})^T A^T,$$

y si consideramos las matrices

$$\widetilde{S}^{\phi} = \sum_{i,k\colon \phi(x_k)\in N^e_{\phi(x_i)}} \frac{(K_{.i}-K_{.k})(K_{.i}-K_{.k})^T}{|N^e_{\phi(x_i)}|}$$
$$\widetilde{C}^{\phi} = \sum_{i,j\colon \phi(x_j)\in N^o_{\phi(x_i)}} \frac{(K_{.i}-K_{.j})(K_{.i}-K_{.j})^T}{|N^o_{\phi(x_i)}|},$$

se cumple que el margen promedio viene dado por

$$\gamma^L = \operatorname{tr}(L(S^{\phi}-C^{\phi})L^T) = \operatorname{tr}(LS^{\phi}L^T - LC^{\phi}L^T) = \operatorname{tr}(A\widetilde{S}^{\phi}A^T - A\widetilde{C}^{\phi}A^T = \operatorname{tr}(A(\widetilde{S}^{\phi}-\widetilde{C}^{\phi})A^T))$$

Si imponemos la restricción $AA^T = I$, el teorema 2.4.6 nos dice de nuevo que podemos tomar como matriz $A$ aquella que contenga por filas los vectores propios de $\widetilde{S}^{\phi} - \widetilde{C}^{\phi}$ asociados a sus $d'$ valores propios. Observemos que ambas matrices podemos calcularlas a partir de la función kernel, y la matriz $A$ así obtenida determina la aplicación lineal, por el teorema de representación. Por tanto, hemos obtenido finalmente un método basado en kernels para la aplicación de ANMM en espacios de características.

### 6.6.4. KDMLMJ

KDMLMJ [25] es la versión kernelizada de DMLMJ. En ella, los datos del conjunto $\mathcal{X}$ se envían al espacio de características mediante la aplicación $\phi\colon \mathcal{X} \to \mathcal{F}$, en el cual se aplica DMLMJ para obtener una aplicación lineal.

De nuevo, es posible calcular los vecindarios $k$-positivo y $k$-negativo $V_k^+(\phi(x_i))$ y $V_k^-(\phi(x_i))$ para cada $x_i \in \mathcal{X}$ gracias a la igualdad 6.39. No ocurre lo mismo con los endomorfismos asociadas a los espacios de diferencias,

$$\Sigma_S^{\phi} = \frac{1}{|S|}\sum_{i=1}^{N}\left[\sum_{\phi(x_j)\in V_k^+(\phi(x_i))}(\phi(x_i)-\phi(x_j))(\phi(x_i)-\phi(x_j))^T\right]$$
$$\Sigma_D^{\phi} = \frac{1}{|D|}\sum_{i=1}^{N}\left[\sum_{\phi(x_j)\in V_k^-(\phi(x_i))}(\phi(x_i)-\phi(x_j))(\phi(x_i)-\phi(x_j))^T\right].$$

El problema de optimización viene dado por

$$\max_{L\in\mathcal{L}(\mathcal{F},\mathbb{R}^{d'})} \quad J(L) = \operatorname{tr}\left((L\Sigma_S^{\phi}L^T)^{-1}(L\Sigma_D^{\phi}L^T) + (L\Sigma_D^{\phi}L^T)^{-1}(L\Sigma_S^{\phi}L^T)\right).$$

De nuevo se tiene, por los teoremas de representación, que $L\phi(x_i) = AK_{.i}$ para cada $x_i \in \mathcal{X}$, donde $A$ es la matriz del teorema de representación y $K_{.i}$ es la $i$-ésima columna de la matriz de kernel. Si, razonando como en la sección anterior, definimos las matrices

$$U = \frac{1}{|S|}\sum_{i=1}^{N}\left[\sum_{\phi(x_j)\in V_k^+(\phi(x_i))}(K_{.i}-K_{.j})(K_{.i}-K_{.j})^T\right]$$
$$V = \frac{1}{|D|}\sum_{i=1}^{N}\left[\sum_{\phi(x_j)\in V_k^-(\phi(x_i))}(K_{.i}-K_{.j})(K_{.i}-K_{.j})^T\right],$$

obtenemos que

$$\operatorname{tr}\left((L\Sigma_S^\phi L^T)^{-1}(L\Sigma_D^\phi L^T) + (L\Sigma_D^\phi L^T)^{-1}(L\Sigma_S^\phi L^T)\right) = \operatorname{tr}\left((AUA^T)^{-1}(AVA^T) + (AVA^T)^{-1}(AUA^T)\right).$$

De forma análoga a DMLMJ, podemos encontrar una matriz $A$ que maximice esta última igualdad tomando los vectores propios de $U^{-1}V$ para los que se maximice el valor $\lambda + 1/\lambda$, donde $\lambda$ es el valor propio asociado. Como las matrices $U$ y $V$ se pueden obtener a partir de la función kernel, y $A$ determina a $L$ por el teorema de representación, hemos obtenido un algoritmo para la aplicación de DMLMJ en el espacio de características.

### 6.6.5. KDA

KDA (*Kernel Discriminant Analysis*) [24] es la versión kernelizada del análisis discriminante lineal. La kernelización de este algoritmo permitirá encontrar direcciones no lineales que separen bien a los datos de acuerdo con el criterio establecido en el análisis discriminante. Una vez más, enviamos los datos del conjunto $\mathcal{X}$ al espacio de características mediante la aplicación $\phi\colon \mathcal{X} \to \mathcal{F}$. Sobre dicho espacio aplicaremos el análisis discriminante lineal.

Supongamos, al igual que en LDA, que el conjunto de posibles clases es $\mathcal{C}$, de cardinal $r$, y para cada $c \in \mathcal{C}$ definimos $\mathcal{C}_c = \{i \in \{1,\dots,N\}\colon y_i = c\}$ y $N_c = |\mathcal{C}_c|$, con $\mu_c^\phi$ el vector media de la clase $c$ y $\mu^\phi$ el vector media de todos los datos, considerándolos dentro del espacio de características. El problema que buscamos resolver en este caso es

$$\max_{L \in \mathcal{L}(\mathcal{F}, \mathbb{R}^{d'})} \quad \operatorname{tr}\left(\frac{LS_b^\phi L^T}{LS_w^\phi L^T}\right), \tag{6.42}$$

donde $S_b^\phi$ y $S_w^\phi$ son los operadores que miden la dispersión entre clases e intra-clase, respectivamente, y vienen dados por

$$S_b^\phi = \sum_{c \in \mathcal{C}} (\mu_c^\phi - \mu^\phi)(\mu_c^\phi - \mu^\phi)^T$$
$$S_w^\phi = \sum_{c \in \mathcal{C}} \sum_{i \in \mathcal{C}_c} (\phi(x_i) - \mu_c^\phi)(\phi(x_i) - \mu_c^\phi)^T,$$

De nuevo hacemos uso de los teoremas de representación, de forma que si $L = (\langle w_1, \cdot\rangle, \dots, \langle w_{d'}, \cdot\rangle)$, se verifica que $w_i = \sum_{j=1}^N \alpha_j^i \phi(x_j)$ para cada $i = 1, \dots, d'$ y

$$L\phi(x) = A \begin{pmatrix} K(x_1, x) \\ \vdots \\ K(x_N, x) \end{pmatrix},$$

para los coeficientes $\alpha_j^i$ y la matriz $A$ en las condiciones del teorema de representación. Vamos a buscar de nuevo una expresión del problema 6.42 que dependa únicamente de la función kernel y de la matriz $A$. Para ello, observemos que para los vectores media de cada clase se verifica

$$L\mu_c^\phi = L\left(\frac{1}{N_c}\sum_{i \in \mathcal{C}_c} \phi(x_i)\right) = \frac{1}{N_c}\sum_{i \in \mathcal{C}_c} L\phi(x_i) = \frac{1}{N_c}\sum_{i \in \mathcal{C}_c} AK_{\cdot i},$$

donde $K_{.i}$ es la columna $i$-ésima de la matriz kernel. Análogamente, para el vector media global, se tiene

$$L\mu^{\phi} = \frac{1}{N}\sum_{i=1}^{N} AK_{.i}.$$

En consecuencia,

$$\begin{aligned} L(\mu_c^{\phi} - \mu^{\phi})(\mu_c^{\phi} - \mu^{\phi})^T L^T &= (L\mu_c^{\phi} - L\mu^{\phi})(L\mu_c^{\phi} - L\mu^{\phi})^T \\ &= \left(\frac{1}{N_c}\sum_{i\in\mathcal{C}_c} AK_{.i} - \frac{1}{N}\sum_{i=1}^{N} AK_{.i}\right)\left(\frac{1}{N_c}\sum_{i\in\mathcal{C}_c} AK_{.i} - \frac{1}{N}\sum_{i=1}^{N} AK_{.i}\right)^T. \end{aligned}$$

Notemos que la última expresión depende únicamente de $A$ y de la función kernel. Por otra parte, para $x_i \in \mathcal{X}$ con $y_i = c$ se tiene

$$\begin{aligned} L(\phi(x_i) - \mu_c^{\phi})(\phi(x_i) - \mu_c^{\phi})^T L^T &= (L\phi(x_i) - L\mu_c^{\phi})(L\phi(x_i) - L\mu_c^{\phi})^T \\ &= \left(AK_{.i} - \frac{1}{N_c}\sum_{j\in\mathcal{C}_c} AK_{.j}\right)\left(AK_{.i} - \frac{1}{N_c}\sum_{j\in\mathcal{C}_c} AK_{.j}\right)^T \\ &= \left(AK_{.i} - \frac{1}{N_c}\sum_{j\in\mathcal{C}_c} AK_{.j}\right)\left(K_{.i}^T A^T - \frac{1}{N_c}\sum_{j\in\mathcal{C}_c} K_{.j}^T A^T\right) \\ &= AK_{.i}K_{.i}^T A^T - \frac{1}{N_c}\sum_{j\in\mathcal{C}_c} AK_{.i}K_{.j}^T A^T - \frac{1}{N_c}\sum_{j\in\mathcal{C}_c} AK_{.j}K_{.i}^T A^T + \frac{1}{N_c^2}\sum_{j\in\mathcal{C}_c}\sum_{l\in\mathcal{C}_c} AK_{.j}K_{.l}^T A^T. \end{aligned}$$

Sumando en $i \in \mathcal{C}_c$ obtenemos

$$\begin{aligned} &\sum_{i\in\mathcal{C}_c} L(\phi(x_i) - \mu_c^{\phi})(\phi(x_i) - \mu_c^{\phi})^T L^T \\ &= \sum_{i\in\mathcal{C}_c}\left[AK_{.i}K_{.i}^T A^T - \frac{1}{N_c}\sum_{j\in\mathcal{C}_c} AK_{.i}K_{.j}^T A^T - \frac{1}{N_c}\sum_{j\in\mathcal{C}_c} AK_{.j}K_{.i}^T A^T + \frac{1}{N_c^2}\sum_{j\in\mathcal{C}_c}\sum_{l\in\mathcal{C}_c} AK_{.j}K_{.l}^T A^T\right] \\ &= \sum_{i\in\mathcal{C}_c} AK_{.i}K_{.i}^T A^T - \frac{2}{N_c}\sum_{i\in\mathcal{C}_c}\sum_{j\in\mathcal{C}_c} AK_{.i}K_{.j}^T A^T + \frac{1}{N_c^2}\sum_{i\in\mathcal{C}_c}\sum_{j\in\mathcal{C}_c}\sum_{l\in\mathcal{C}_c} AK_{.j}K_{.l}^T A^T \\ &= \sum_{i\in\mathcal{C}_c} AK_{.i}K_{.i}^T A^T - \frac{2}{N_c}\sum_{i\in\mathcal{C}_c}\sum_{j\in\mathcal{C}_c} AK_{.i}K_{.j}^T A^T + \frac{N_c}{N_c^2}\sum_{j\in\mathcal{C}_c}\sum_{l\in\mathcal{C}_c} AK_{.j}K_{.l}^T A^T \\ &= \sum_{i\in\mathcal{C}_c} AK_{.i}K_{.i}^T A^T - \frac{1}{N_c}\sum_{i\in\mathcal{C}_c}\sum_{j\in\mathcal{C}_c} AK_{.i}K_{.j}^T A^T \\ &= AK_cK_c^T A^T - AK_c\left(\frac{1}{N_c}\mathbb{1}\right)K_c^T A^T \\ &= AK_c\left(I - \frac{1}{N_c}\mathbb{1}\right)K_c^T A^T, \end{aligned}$$

donde $\mathbb{1} \in \mathcal{M}_{N_c}(\mathbb{R})$ es una matriz cuadrada con todos sus términos de valor 1 y $K_c \in \mathcal{M}_{N\times N_c}$ tiene como entradas los valores de la función kernel entre todos los elementos de $\mathcal{X}$ y los elementos de clase $c$. De nuevo, esta última expresión solo depende de $A$ y de la función kernel.

Si finalmente definimos

$$U_c = \frac{1}{N_c} \sum_{i \in \mathcal{C}_c} K_{.i} \in \mathbb{R}^N, c \in \mathcal{C}$$

$$U_\mu = \frac{1}{N} \sum_{j=1}^{N} K_{.i} \in \mathbb{R}^N$$

$$U = \sum_{c \in \mathcal{C}} N_c (U_c - U_\mu)(U_c - U_\mu)^T \in S_N(\mathbb{R})$$

$$V = \sum_{c \in \mathcal{C}} K_c \left( I - \frac{1}{N_c} \mathbb{1} \right) K_c^T \in S_N(\mathbb{R}),$$

se concluye que

$$\operatorname{tr}\left( \frac{L S_b^\phi L^T}{L S_w^\phi L^T} \right) = \operatorname{tr}\left( \frac{AUA^T}{AVA^T} \right),$$

donde $U$ y $V$ son calculables a partir de funciones kernel. Por tanto, obtenemos un problema equivalente al original en términos de $A$, para el cual el teorema 2.4.8 nos dice que, si $U$ es definida positiva, podemos maximizar el valor de la traza tomando como filas de $A$ los vectores propios de $U^{-1}V$ asociados a sus $d'$ mayores valores propios. De esta forma, puesto que $A$ determina a $L$ gracias al teorema de representación, obtenemos un método basado en kernels para la aplicación del análisis discriminante en espacios de características.

# Parte III

# Informática práctica

# Capítulo 7

# Software desarrollado

## 7.1. Los lenguajes R y Python

## 7.2. Descripción del software

## 7.3. Uso del software

# Capítulo 8

# Experimentación

## 8.1. Descripción de los experimentos

## 8.2. Resultados

## 8.3. Conclusiones

# Bibliografía


[1] Yaser S Abu-Mostafa, Malik Magdon-Ismail y Hsuan-Tien Lin. *Learning from data*. Vol. 4. AMLBook New York, NY, USA: 2012.

[2] Aurelien Bellet, Amaury Habrard y Marc Sebban. "A survey on metric learning for feature vectors and structured data". En: *arXiv preprint arXiv:1306.6709* (2013).

[3] Stephen Boyd y Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[4] Lev M Bregman. "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming". En: *USSR computational mathematics and mathematical physics* 7.3 (1967), págs. 200-217.

[5] Ratthachat Chatpatanasiri y col. "A new kernelization framework for Mahalanobis distance learning algorithms". En: *Neurocomputing* 73.10-12 (2010), págs. 1570-1579.

[6] Ward Cheney y Allen A Goldstein. "Proximity maps for convex sets". En: *Proceedings of the American Mathematical Society* 10.3 (1959), págs. 448-450.

[7] Vladimir Cherkassky y Filip Mulier. *Learning from data: Concepts, theory, and methods*. Wiley New York, 1998.

[8] Thomas H Cormen. *Introduction to algorithms*. MIT press, 2009.

[9] Thomas M. Cover y Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. New York, NY, USA: Wiley-Interscience, 2006. ISBN: 0471241954.

[10] Bernard Dacorogna. *Direct methods in the calculus of variations*. Vol. 78. Springer Science & Business Media, 2007, págs. 67-148.

[11] Jason V Davis y Inderjit S Dhillon. "Differential entropic clustering of multivariate gaussians". En: *Advances in Neural Information Processing Systems*. 2007, págs. 337-344.

[12] Jason V Davis y col. "Information-theoretic metric learning". En: *Proceedings of the 24th international conference on Machine learning*. ACM. 2007, págs. 209-216.

[13] Mikel Galar y col. "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes". En: *Pattern Recognition* 44.8 (2011), págs. 1761-1776.

[14] Amir Globerson y Sam T Roweis. "Metric learning by collapsing classes". En: *Advances in neural information processing systems*. 2006, págs. 451-458.

[15] Jacob Goldberger y col. "Neighbourhood components analysis". En: *Advances in neural information processing systems*. 2005, págs. 513-520.

[16] Matthieu Guillaumin, Jakob Verbeek y Cordelia Schmid. "Is that you? Metric learning approaches for face identification". En: *Computer Vision, 2009 IEEE 12th international conference on*. IEEE. 2009, págs. 498-505.

[17] Nicholas J Higham. "Computing a nearest symmetric positive semidefinite matrix". En: *Linear algebra and its applications* 103 (1988), págs. 103-118.

[18] Thomas Hofmann, Bernhard Schölkopf y Alexander J Smola. "Kernel methods in machine learning". En: *The annals of statistics* (2008), págs. 1171-1220.

[19] Roger A Horn, Roger A Horn y Charles R Johnson. *Matrix analysis*. Cambridge university press, 1990.

[20] Gareth James y col. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[21] Peter Lancaster y Miron Tismenetsky. *The theory of matrices: with applications*. Elsevier, 1985.

[22] Ujjwal Maulik y Sanghamitra Bandyopadhyay. "Performance evaluation of some clustering algorithms and validity indices". En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.12 (2002), págs. 1650-1654.

[23] Thomas Mensink y col. "Metric learning for large scale image classification: Generalizing to new classes at near-zero cost". En: *Computer Vision–ECCV 2012*. Springer, 2012, págs. 488-501.

[24] Sebastian Mika y col. "Fisher discriminant analysis with kernels". En: *Neural networks for signal processing IX, 1999. Proceedings of the 1999 IEEE signal processing society workshop.* Ieee. 1999, págs. 41-48.

[25] Bac Nguyen, Carlos Morell y Bernard De Baets. "Supervised distance metric learning through maximization of the Jeffrey divergence". En: *Pattern Recognition* 64 (2017), págs. 215-225.

[26] Michael L Overton. "On minimizing the maximum eigenvalue of a symmetric matrix". En: *SIAM Journal on Matrix Analysis and Applications* 9.2 (1988), págs. 256-268.

[27] Kaare Brandt Petersen, Michael Syskind Pedersen y col. "The matrix cookbook". En: *Technical University of Denmark* 7.15 (2008), pág. 510.

[28] Ralph Tyrell Rockafellar. *Convex analysis*. Princeton university press, 2015.

[29] Walter Rudin. *Real and complex analysis*. Tata McGraw-Hill Education, 1987.

[30] Bernhard Schölkopf, Alexander Smola y Klaus-Robert Müller. "Nonlinear component analysis as a kernel eigenvalue problem". En: *Neural computation* 10.5 (1998), págs. 1299-1319.

[31] Shai Shalev-Shwartz y Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[32] Amar Subramanya y Partha Pratim Talukdar. "Graph-based semi-supervised learning algorithms for NLP". En: *Tutorial Abstracts of ACL 2012*. Association for Computational Linguistics. 2012, págs. 6-6.

[33] Robert Thompson. "Convex and concave functions of singular values of matrix sums". En: *Pacific Journal of Mathematics* 66.1 (1976), págs. 285-290.

[34] Lorenzo Torresani y Kuang-chih Lee. "Large margin component analysis". En: *Advances in neural information processing systems*. 2007, págs. 1385-1392.

[35] Fei Wang y Jimeng Sun. "Survey on distance metric learning and dimensionality reduction in data mining". En: *Data Mining and Knowledge Discovery* 29.2 (2015), págs. 534-564.

[36] Fei Wang y Changshui Zhang. "Feature extraction by maximizing the average neighborhood margin". En: *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE. 2007, págs. 1-8.

[37] Kilian Q Weinberger y Lawrence K Saul. "Distance metric learning for large margin nearest neighbor classification". En: *Journal of Machine Learning Research* 10.Feb (2009), págs. 207-244.

[38] Eric P Xing y col. "Distance metric learning with application to clustering with side-information". En: *Advances in neural information processing systems*. 2003, págs. 521-528.

[39] Rui Xu y Donald Wunsch. "Survey of clustering algorithms". En: *IEEE Transactions on neural networks* 16.3 (2005), págs. 645-678.

[40] Yiming Ying y Peng Li. "Distance metric learning with eigenvalue optimization". En: *Journal of Machine Learning Research* 13.Jan (2012), págs. 1-26.

[41] Fuzhen Zhang. *Matrix theory: basic results and techniques*. Springer Science & Business Media, 2011.

[42] Xiaojin Zhu y Zoubin Ghahramani. "Learning from labeled and unlabeled data with label propagation". En: (2002).