# Tarea 1

Aprendizaje supervisado, funciones de pérdida y dimensión VC
Reconocimiento de Patrones y AA 26-01

**Instrucciones:** Resuelve los siguientes ejercicios de manera individual. La entrega es en un documento PDF. Puedes escribir tus respuestas en el documento mediante algún software o completamente a mano. Si lo haces a mano, cada paso debe ser legible, y las fotos que tomes se deben incorporar como parte del documento PDF.

## 1. [2.5 puntos] Dimensión VC

Para una clase de hipótesis de triángulos $\mathcal{H}_{\text{tri}}$:

a) Escribe una función indicadora para evaluar si un punto 2D pertenece o no a una hipótesis de un triángulo particular $h_{\boldsymbol{\theta}} \in \mathcal{H}_{\text{tri}}$.

   La función indicadora debe determinar si un punto $(x, y) \in \mathbb{R}^2$ está dentro o fuera de un triángulo, regresando 1 si el punto está dentro y 0 si está fuera.

   Utilizo las coordenadas baricéntricas para resolver este problema. Para un triángulo con vértices $V_1 = (x_1, y_1)$, $V_2 = (x_2, y_2)$, $V_3 = (x_3, y_3)$, cualquier punto $P = (x, y)$ se puede expresar como:

$$P = \lambda_1 V_1 + \lambda_2 V_2 + \lambda_3 V_3$$

   donde $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

   Las coordenadas $\lambda_1$, $\lambda_2$ y $\lambda_3$ se obtienen resolviendo el sistema:

$$x = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 \tag{1}$$
$$y = \lambda_1 y_1 + \lambda_2 y_2 + \lambda_3 y_3 \tag{2}$$
$$1 = \lambda_1 + \lambda_2 + \lambda_3 \tag{3}$$

   La solución es:

$$\lambda_1 = \frac{(y_2 - y_3)(x - x_3) + (x_3 - x_2)(y - y_3)}{(y_2 - y_3)(x_1 - x_3) + (x_3 - x_2)(y_1 - y_3)}$$

$$\lambda_2 = \frac{(y_3 - y_1)(x - x_3) + (x_1 - x_3)(y - y_3)}{(y_2 - y_3)(x_1 - x_3) + (x_3 - x_2)(y_1 - y_3)}$$

$$\lambda_3 = 1 - \lambda_1 - \lambda_2$$

**Criterio de pertenencia:**

Un punto pertenece al triángulo si y solo si todas sus coordenadas baricéntricas son no negativas:

$$P \in \text{Triángulo} \Leftrightarrow \lambda_1 \geq 0 \wedge \lambda_2 \geq 0 \wedge \lambda_3 \geq 0$$

**Función indicadora resultante:**

$$h_{\boldsymbol{\theta}}(x, y) = \begin{cases} 1 & \text{si } \lambda_1 \geq 0 \wedge \lambda_2 \geq 0 \wedge \lambda_3 \geq 0 \\ 0 & \text{en caso contrario} \end{cases}$$

b) Debes escribir la función parametrizada por los parámetros $\boldsymbol{\theta}$ que tú consideres.

**Parametrización propuesta:**

Un triángulo en $\mathbb{R}^2$ queda completamente determinado por sus tres vértices. Cada vértice tiene dos coordenadas $(x_i, y_i)$, por lo que defino 6 parámetros en total.

**Vector de parámetros:**

$$\boldsymbol{\theta} = (x_1, y_1, x_2, y_2, x_3, y_3) \in \mathbb{R}^6$$

**Función parametrizada:**

La función $h_{\boldsymbol{\theta}} : \mathbb{R}^2 \to \{0, 1\}$ se define como:

$$h_{\boldsymbol{\theta}}(x, y) = \mathbb{1}_{\{\lambda_1(\boldsymbol{\theta}, x, y) \geq 0\}} \cdot \mathbb{1}_{\{\lambda_2(\boldsymbol{\theta}, x, y) \geq 0\}} \cdot \mathbb{1}_{\{\lambda_3(\boldsymbol{\theta}, x, y) \geq 0\}}$$

donde $\mathbb{1}_{\{A\}}$ es la función indicadora del evento $A$.

**Implementación explícita:**

Definiendo el denominador común:

$$D(\boldsymbol{\theta}) = (y_2 - y_3)(x_1 - x_3) + (x_3 - x_2)(y_1 - y_3)$$

Si $|D(\boldsymbol{\theta})| > \epsilon$ (triángulo no degenerado), entonces:

$$h_{\boldsymbol{\theta}}(x, y) = \begin{cases} & \dfrac{(y_2 - y_3)(x - x_3) + (x_3 - x_2)(y - y_3)}{D(\boldsymbol{\theta})} \geq 0 \text{ y} \\ 1 & \text{si } \dfrac{(y_3 - y_1)(x - x_3) + (x_1 - x_3)(y - y_3)}{D(\boldsymbol{\theta})} \geq 0 \text{ y} \\ & 1 - \lambda_1 - \lambda_2 \geq 0 \\ 0 & \text{en caso contrario} \end{cases}$$

Si $|D(\boldsymbol{\theta})| \leq \epsilon$, entonces $h_{\boldsymbol{\theta}}(x, y) = 0$ (triángulo degenerado).

## 2. [2.5 puntos] Clases de hipótesis

Para una clase de hipótesis de elipses $\mathcal{H}_{\text{eli}}$ donde una elipse con centro en la coordenada $(h, k)$ se describe con la siguiente ecuación:

$$\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} = 1 \tag{4}$$

donde $a > 0$ y $b > 0$ son los parámetros para los semiejes de la elipse.

a) Usa esta ecuación para escribir una función indicadora para evaluar si un punto en 2D pertenece o no a una hipótesis de una elipse particular $h_{\boldsymbol{\theta}} \in \mathcal{H}_{\text{eli}}$, donde $\boldsymbol{\theta} = \{h, k, a, b\}$.

**Solución:**

La ecuación estándar de una elipse con centro en $(h, k)$ y semiejes $a$ y $b$ es:

$$\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} = 1$$

Esta ecuación define la frontera de la elipse. Para la función indicadora, generalizo para incluir el interior:

**Criterio de pertenencia:**

Defino la distancia normalizada:

$$d(x, y) = \frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2}$$

La interpretación es:

- Si $d(x, y) < 1$: el punto está dentro de la elipse
- Si $d(x, y) = 1$: el punto está sobre la elipse
- Si $d(x, y) > 1$: el punto está fuera de la elipse

**Función indicadora:**

Para clasificación binaria, considero que un punto pertenece a la clase positiva si está dentro o sobre la elipse:

$$h_{\boldsymbol{\theta}}(x, y) = \begin{cases} 1 & \text{si } \frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} \leq 1 \\ 0 & \text{en caso contrario} \end{cases}$$

La función indicadora $h_{\boldsymbol{\theta}} : \mathbb{R}^2 \to \{0, 1\}$ divide el plano en dos regiones: el interior (incluyendo la frontera) de la elipse y el exterior.

b) Escribe a manera de pseudocódigo un algoritmo para encontrar los parámetros $\boldsymbol{\theta}' = \{h', k', a', b'\}$ de la elipse más pequeña que sólo permite un error menor o igual a un umbral $\alpha$ (relativamente pequeño; por ejemplo, 10 %) de clasificaciones incorrectas (para una de dos clases) del total de puntos $\mathcal{X} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$.

**Solución:**

**Formulación del problema**

Buscamos la elipse más pequeña (en área) que clasifique correctamente al menos $(1-\alpha)\cdot n$ puntos, donde $n = |\mathcal{X}|$.

**Definición matemática:** Minimizar $\pi a'b'$ (área de la elipse) sujeto a:

$$\sum_{i=1}^{n} \nVdash_{\{h_{\boldsymbol{\theta}'}(x_i,y_i)\neq y_i\}} \leq \lfloor \alpha \cdot n \rfloor$$

**Estrategia de escalamiento**

La idea es:

*a*) Estimar la forma y orientación de la elipse usando la clase positiva

*b*) Encontrar la escala mínima que respete el presupuesto de error

**Algoritmo paso a paso**

**Algorithm 1** EllipseMinima - Encontrar elipse mínima con umbral de error

**Require:** $\mathcal{X} = \{(x_i, y_i)\}_{i=1}^{n}$, $\mathcal{Y} = \{y_i\}_{i=1}^{n} \subseteq \{-1, +1\}$, $\alpha \in (0, 1)$
**Ensure:** $\boldsymbol{\theta}' = \{h', k', a', b'\}$

$\triangleright$ Separación por clases
$\mathcal{X}^+ \leftarrow \{(x_i, y_i) : y_i = +1\}$
$\mathcal{X}^- \leftarrow \{(x_i, y_i) : y_i = -1\}$
$n^+ \leftarrow |\mathcal{X}^+|$, $n^- \leftarrow |\mathcal{X}^-|$
max_errores $\leftarrow \lfloor \alpha \cdot n \rfloor$

$\triangleright$ Estimación del centro
$h' \leftarrow \frac{1}{n^+} \sum_{(x,y)\in\mathcal{X}^+} x$
$k' \leftarrow \frac{1}{n^+} \sum_{(x,y)\in\mathcal{X}^+} y$

$\triangleright$ Estimación de la forma base
$\epsilon \leftarrow 1 \times 10^{-9}$
$a_0 \leftarrow \sqrt{\frac{1}{n^+} \sum_{(x,y)\in\mathcal{X}^+} (x - h')^2} + \epsilon$
$b_0 \leftarrow \sqrt{\frac{1}{n^+} \sum_{(x,y)\in\mathcal{X}^+} (y - k')^2} + \epsilon$

$\triangleright$ Cálculo de distancias normalizadas
**for** cada punto $(x_i, y_i) \in \mathcal{X}$ **do**
    $d_i \leftarrow \sqrt{\frac{(x_i - h')^2}{a_0^2} + \frac{(y_i - k')^2}{b_0^2}}$
**end for**

$\triangleright$ Búsqueda de la escala óptima
$\mathcal{S} \leftarrow$ valores únicos ordenados de $\{d_i : i = 1, \ldots, n\}$
**for** cada $s \in \mathcal{S}$ (en orden ascendente) **do**
    errores $\leftarrow 0$
    **for** $i = 1$ **to** $n$ **do**
        **if** $d_i \leq s$ **then**
            $\text{pred}_i \leftarrow +1$
        **else**
            $\text{pred}_i \leftarrow -1$
        **end if**
        **if** $\text{pred}_i \neq y_i$ **then**
            errores $\leftarrow$ errores $+ 1$
        **end if**
    **end for**
    **if** errores $\leq$ max_errores **then**
        $s^* \leftarrow s$
        **break**
    **end if**
**end for**

$\triangleright$ Parámetros finales
$a' \leftarrow a_0 \cdot s^*$
$b' \leftarrow b_0 \cdot s^*$
**return** $\boldsymbol{\theta}' = \{h', k', a', b'\}$

c) Ahora encuentra también a manera de pseudocódigo los parámetros del elipse más grande que no introduce nuevos errores. Puedes partir de los parámetros $\boldsymbol{\theta}'$ del inciso anterior.

**Solución:**

**Problema de optimización**

Dada la elipse mínima $\boldsymbol{\theta}' = \{h', k', a', b'\}$, buscamos la elipse máxima $\boldsymbol{\theta}'' = \{h', k', a'', b''\}$ tal que:

$$\text{errores}(\boldsymbol{\theta}'') = \text{errores}(\boldsymbol{\theta}')$$

Es decir, no introducimos errores adicionales.

**Restricción geométrica**

Para no introducir nuevos errores, la elipse expandida no debe incluir puntos negativos que estaban correctamente clasificados como negativos.

**Criterio:** Si $(x_i, y_i) \in \mathcal{X}^-$ y está correctamente clasificado por $\boldsymbol{\theta}'$, entonces:

$$\frac{(x_i - h')^2}{(a')^2} + \frac{(y_i - k')^2}{(b')^2} > 1$$

**Factor de expansión**

Si escalamos uniformemente: $a'' = s \cdot a'$ y $b'' = s \cdot b'$ con $s \geq 1$, entonces la nueva distancia normalizada para el punto $(x_i, y_i)$ es:

$$d_i^{(s)} = \sqrt{\frac{(x_i - h')^2}{(s \cdot a')^2} + \frac{(y_i - k')^2}{(s \cdot b')^2}} = \frac{d_i^{(\text{original})}}{s}$$

**Algoritmo de expansión**

**Algorithm 2** ElipseMaxima - Encontrar elipse máxima sin nuevos errores

**Require:** $\boldsymbol{\theta}' = \{h', k', a', b'\}$, $\mathcal{X}$, $\mathcal{Y}$
**Ensure:** $\boldsymbol{\theta}'' = \{h', k', a'', b''\}$

▷ Calcular distancias originales
**for** cada punto $(x_i, y_i) \in \mathcal{X}$ **do**
  $d_i^{(\text{original})} \leftarrow \sqrt{\frac{(x_i-h')^2}{(a')^2} + \frac{(y_i-k')^2}{(b')^2}}$
**end for**

▷ Identificar puntos negativos externos
$\mathcal{N} \leftarrow \{i : y_i = -1 \text{ y } d_i^{(\text{original})} > 1\}$

▷ Calcular factor de expansión máximo
**if** $\mathcal{N} \neq \emptyset$ **then**
  $\delta \leftarrow 1 \times 10^{-6}$
  $s_{\text{máx}} \leftarrow \text{mín}_{i\in\mathcal{N}}\, d_i^{(\text{original})} - \delta$
  $s_{\text{máx}} \leftarrow \text{máx}(s_{\text{máx}}, 1{,}0)$
**else**
  $s_{\text{máx}} \leftarrow 1{,}0$
**end if**

▷ Parámetros de la elipse máxima
$a'' \leftarrow s_{\text{máx}} \cdot a'$
$b'' \leftarrow s_{\text{máx}} \cdot b'$

▷ Verificación final
errores_original $\leftarrow$ contar_errores($\mathcal{X}$, $\mathcal{Y}$, $\boldsymbol{\theta}'$)
errores_expandida $\leftarrow$ contar_errores($\mathcal{X}$, $\mathcal{Y}$, $\{h', k', a'', b''\}$)
**while** errores_expandida > errores_original **do**
  $s_{\text{máx}} \leftarrow s_{\text{máx}} \times 0{,}999$
  $a'' \leftarrow s_{\text{máx}} \cdot a'$
  $b'' \leftarrow s_{\text{máx}} \cdot b'$
  errores_expandida $\leftarrow$ contar_errores($\mathcal{X}$, $\mathcal{Y}$, $\{h', k', a'', b''\}$)
**end while**

**return** $\boldsymbol{\theta}'' = \{h', k', a'', b''\}$

d) Finalmente propón como hipótesis para la función indicadora un elipse que esté entre ambos (de los incisos b y c), por ejemplo a la mitad del camino paramétricamente hablando.

**Solución:**

**Motivación para la interpolación**

La elipse mínima $\boldsymbol{\theta}'$ minimiza el error de entrenamiento pero puede ser muy conservadora (alto sesgo). La elipse máxima $\boldsymbol{\theta}''$ maximiza el margen pero puede ser muy permisiva. Una interpolación puede balancear estos extremos.

**Interpolación lineal**

Para un parámetro de interpolación $\lambda \in [0, 1]$, definimos:

$$h_{\text{final}} = h' \quad \text{(el centro no cambia)} \tag{5}$$
$$k_{\text{final}} = k' \quad \text{(el centro no cambia)} \tag{6}$$
$$a_{\text{final}} = (1-\lambda)a' + \lambda a'' \tag{7}$$
$$b_{\text{final}} = (1-\lambda)b' + \lambda b'' \tag{8}$$

**Caso especial ($\lambda = 0{,}5$):** Para el punto medio:

$$a_{\text{final}} = \frac{a' + a''}{2} \tag{9}$$
$$b_{\text{final}} = \frac{b' + b''}{2} \tag{10}$$

**Función indicadora interpolada**

$$h_{\text{final}}(x,y) = \begin{cases} 1 & \text{si } \frac{(x-h')^2}{a_{\text{final}}^2} + \frac{(y-k')^2}{b_{\text{final}}^2} \leq 1 \\ 0 & \text{en caso contrario} \end{cases}$$

**Interpretación geométrica**

La elipse interpolada tiene:

- **Área:** $\pi a_{\text{final}} b_{\text{final}} = \pi \cdot \frac{a'+a''}{2} \cdot \frac{b'+b''}{2}$
- **Error esperado:** Entre el error mínimo y el error de la elipse máxima
- **Generalización:** Potencialmente mejor que los extremos

**Validación**

Para validar la elección de $\lambda$, se puede:

a) Usar validación cruzada para seleccionar $\lambda^* \in [0,1]$
b) Evaluar el error en un conjunto de prueba
c) Comparar con $\lambda = 0$ (elipse mínima) y $\lambda = 1$ (elipse máxima)

e) Implementarás esto en la práctica asociada a este ejercicio.

**Guía para la implementación:**

La implementación debe incluir las siguientes funciones matemáticas:

**Función 1: Evaluación de la función indicadora**

$$f_{\text{indicadora}}(\boldsymbol{\theta}, x, y) = \mathbb{1}_{\left\{\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} \leq 1\right\}}$$

**Función 2: Cálculo de error**

$$E(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{h_{\boldsymbol{\theta}}(x_i,y_i) \neq y_i\}}$$

**Función 3: Optimización de parámetros**

Implementar los algoritmos ElipseMinima y ElipseMaxima descritos en los incisos anteriores.

## 3. [2.5 puntos] Función de pérdida esperada

Dados los valores de una función de pérdida $L_{kj}$, el riesgo esperado

$$E[L] = \sum_k \sum_j \int_{R_j} L_{kj} p(x, \mathcal{C}_k) dx \tag{11}$$

se minimiza si para cada $x$ escogemos la clase para minimizar la siguiente expresión (ya que $p(x)$ no contiene al parámetro de clase):

$$\arg\min_j \left( \sum_k L_{kj} p(\mathcal{C}_k|x) p(x) \right) = \arg\min_j \left( \sum_k L_{kj} p(\mathcal{C}_k|x) \right) \tag{12}$$

Verifica que si $L_{kj} = 1 - I_{kj}$, donde $I_{kj}$ son los elementos de la matriz identidad, entonces la minimización anterior se reduce a escoger la clase $i$ con la probabilidad más grande $p(\mathcal{C}_i|x)$.

**Solución:**

**Desarrollo de la demostración:**

Analizo un problema de decisión bayesiana donde:

- $\mathcal{C}_k$ representa las clases verdaderas ($k = 1, 2, \ldots, K$)
- $j$ representa las decisiones o predicciones ($j = 1, 2, \ldots, K$)
- $L_{kj}$ es la pérdida incurrida al decidir la clase $j$ cuando la clase verdadera es $k$
- $p(\mathcal{C}_k|x)$ es la probabilidad a posteriori de la clase $k$ dado el input $x$

**Definición de la función de pérdida 0-1:**

La función de pérdida $L_{kj} = 1 - I_{kj}$ donde $I_{kj}$ es el elemento $(k, j)$ de la matriz identidad se expresa como:

$$L_{kj} = 1 - I_{kj} = \begin{cases} 0 & \text{si } k = j \text{ (clasificación correcta)} \\ 1 & \text{si } k \neq j \text{ (clasificación incorrecta)} \end{cases}$$

Esta es la función de pérdida 0-1 estándar que penaliza igualmente todos los errores de clasificación.

**Calcular el riesgo esperado para una decisión específica**

Para una decisión fija $j$, el riesgo esperado es:

$$R(j|x) = \sum_k L_{kj} p(\mathcal{C}_k|x)$$

Sustituyendo $L_{kj} = 1 - I_{kj}$:

$$R(j|x) = \sum_k (1 - I_{kj}) p(\mathcal{C}_k|x) \tag{13}$$

$$= \sum_k p(\mathcal{C}_k|x) - \sum_k I_{kj} p(\mathcal{C}_k|x) \tag{14}$$

**Simplificar usando propiedades de las probabilidades**

Analizamos cada término:

**Primer término:** $\sum_k p(\mathcal{C}_k|x) = 1$ (axioma de probabilidad)
**Segundo término:** $\sum_k I_{kj} p(\mathcal{C}_k|x)$
Por definición de la matriz identidad, $I_{kj} = 1$ solo cuando $k = j$, y $I_{kj} = 0$ en todos los demás casos. Por tanto:

$$\sum_k I_{kj} p(\mathcal{C}_k|x) = I_{jj} p(\mathcal{C}_j|x) + \sum_{k \neq j} I_{kj} p(\mathcal{C}_k|x) = 1 \cdot p(\mathcal{C}_j|x) + \sum_{k \neq j} 0 \cdot p(\mathcal{C}_k|x) = p(\mathcal{C}_j|x)$$

**Expresión final del riesgo**
Combinando los resultados:

$$R(j|x) = 1 - p(\mathcal{C}_j|x)$$

**Interpretación:** El riesgo de decidir la clase $j$ es igual a la probabilidad de que la clase verdadera NO sea $j$.
**Encontrar la decisión óptima**
Para minimizar el riesgo esperado, debemos encontrar:

$$j^* = \arg\min_j R(j|x) = \arg\min_j [1 - p(\mathcal{C}_j|x)]$$

Como el término constante 1 no afecta la minimización:

$$j^* = \arg\min_j [-p(\mathcal{C}_j|x)] = \arg\max_j p(\mathcal{C}_j|x)$$

**Conclusión y interpretación**
**Resultado principal:** Con la función de pérdida 0-1, la regla de decisión óptima de Bayes es:

$$i^* = \arg\max_j p(\mathcal{C}_j|x)$$

**Interpretación práctica:**

- Elegir siempre la clase con mayor probabilidad a posteriori
- Esta regla minimiza la probabilidad de error de clasificación
- Es la base teórica para muchos clasificadores bayesianos

**Verificación matemática**
Para verificar que efectivamente hemos demostrado lo requerido, observemos que:

1. **Partimos de:** $\arg\min_j \left(\sum_k L_{kj} p(\mathcal{C}_k|x)\right)$ con $L_{kj} = 1 - I_{kj}$
2. **Demostramos que esto es igual a:** $\arg\min_j [1 - p(\mathcal{C}_j|x)]$
3. **Lo cual es equivalente a:** $\arg\max_j p(\mathcal{C}_j|x)$
4. **Por tanto:** La minimización se reduce efectivamente a escoger la clase $i$ con la probabilidad más grande $p(\mathcal{C}_i|x)$ $\blacksquare$

**Extensión teórica:**
Este resultado es fundamental en la teoría de decisiones bayesianas porque establece que, bajo pérdida 0-1:

- La frontera de decisión óptima está dada por $p(\mathcal{C}_i|x) = p(\mathcal{C}_j|x)$ para clases $i \neq j$
- El error de Bayes (error mínimo alcanzable) es $1 - \max_j p(\mathcal{C}_j|x)$
- Cualquier clasificador que no siga esta regla tendrá error subóptimo

# 4. [2.5 puntos] Función de pérdida relajada

En los ejercicios 1 y 2 obtuviste funciones indicadoras para evaluar la pertenencia o no de puntos a hipótesis concretas.

a) Debes extender la función indicadora para crear una nueva función que no evalúe de manera estrictamente "dura" la pertenencia a la hipótesis sino que permita considerar puntos cercanos a la frontera de la hipótesis de manera penalizada.

**Solución:**

**Justificación para la función suave:**

Las funciones indicadoras binarias presentan limitaciones importantes:

- **Discontinuidad:** Cambio abrupto de 0 a 1 en la frontera
- **No diferenciabilidad:** Dificulta la optimización con métodos basados en gradiente
- **Rigidez:** No permite grados de pertenencia o incertidumbre

**Propuesta de función de pérdida relajada:**

Mi función de pérdida relajada $L_{\text{soft}} : \mathbb{R}^2 \to [0, 1]$ satisface:

*a)* **Continuidad:** $L_{\text{soft}}(x, y)$ es continua

*b)* **Monotonicidad:** Mayor distancia a la frontera implica mayor pérdida

*c)* **Compatibilidad:** $\lim_{\delta \to 0^+} L_{\text{soft}}(x, y) = L_{\text{hard}}(x, y)$

**Definición matemática para elipse**

Para una hipótesis de elipse con parámetros $\boldsymbol{\theta} = \{h, k, a, b\}$, definimos la **distancia normalizada**:

$$d(x, y; \boldsymbol{\theta}) = \sqrt{\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2}}$$

**Interpretación geométrica:**

- $d(x, y) = 1$: punto en la frontera de la elipse
- $d(x, y) < 1$: punto en el interior
- $d(x, y) > 1$: punto en el exterior

**Función de pérdida relajada por partes**

Definimos la función de pérdida suave con parámetro de suavidad $\delta > 0$:

$$L_{\text{soft}}(x, y; \boldsymbol{\theta}, \delta) = \begin{cases} 0 & \text{si } d(x, y) \leq 1 \\ \frac{d(x,y)-1}{\delta} & \text{si } 1 < d(x, y) \leq 1 + \delta \\ 1 & \text{si } d(x, y) > 1 + \delta \end{cases}$$

**Propiedades matemáticas:**

*a)* **Rango:** $L_{\text{soft}} \in [0, 1]$

b) **Continuidad:** La función es continua en todo $\mathbb{R}^2$

c) **Límite:** $\lim_{\delta \to 0^+} L_{\text{soft}}(x, y) = \nVdash_{\{d(x,y)>1\}}$

b) Tú debes definir una manera de extender la frontera hasta cierta distancia máxima más allá de la frontera original. Por ejemplo: englobando a la hipótesis dentro de un círculo con cierto radio, o definiendo una distancia $\Delta d$ más allá de la frontera de la hipótesis.

**Solución:**

**Concepto de zona de transición**

La extensión de la frontera crea una **zona de transición** (o margen suave) alrededor de la hipótesis original donde la clasificación no es binaria sino gradual.

**Definición formal de la extensión**

Para una elipse con parámetros $\boldsymbol{\theta} = \{h, k, a, b\}$ y parámetro de extensión $\delta > 0$, definimos tres regiones:

$$\mathcal{R}_{\text{interior}} = \{(x, y) : d(x, y; \boldsymbol{\theta}) \leq 1\} \tag{15}$$

$$\mathcal{R}_{\text{transición}} = \{(x, y) : 1 < d(x, y; \boldsymbol{\theta}) \leq 1 + \delta\} \tag{16}$$

$$\mathcal{R}_{\text{exterior}} = \{(x, y) : d(x, y; \boldsymbol{\theta}) > 1 + \delta\} \tag{17}$$

**Interpretación geométrica de la extensión**

La zona de transición $\mathcal{R}_{\text{transición}}$ forma una "cáscara elíptica" definida por:

**Frontera interior:** $\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} = 1$

**Frontera exterior:** $\frac{(x-h)^2}{(a\sqrt{1+\delta})^2} + \frac{(y-k)^2}{(b\sqrt{1+\delta})^2} = 1$

**Parametrización alternativa usando distancia euclidiana**

**Método 1: Extensión uniforme**

Extender la elipse uniformemente en todas las direcciones:

$$\boldsymbol{\theta}_{\text{ext}} = \{h, k, a + \Delta a, b + \Delta b\}$$

donde $\Delta a, \Delta b > 0$ son extensiones en los ejes principales.

**Método 2: Extensión por factor de escala**

Escalar la elipse por un factor $s > 1$:

$$\boldsymbol{\theta}_{\text{ext}} = \{h, k, s \cdot a, s \cdot b\}$$

La zona de transición está entre las elipses con factores de escala 1 y $s$.

**Elección del parámetro de extensión**

El parámetro $\delta$ controla el ancho de la zona de transición:

- $\delta$ **pequeño** ($\delta < 0{,}1$)**:** Transición muy estrecha, comportamiento casi binario
- $\delta$ **moderado** ($0{,}1 \leq \delta \leq 1$)**:** Balance entre suavidad y precisión

- $\delta$ **grande ($\delta > 1$):** Transición muy gradual, alta tolerancia

**Criterio de selección:** $\delta$ puede elegirse mediante validación cruzada o como fracción del tamaño característico de la hipótesis (ej. $\delta = 0{,}1 \cdot \text{máx}(a, b)$).

c) En la región donde extiendas la frontera la función de pérdida debe penalizar en lugar de evaluar a cero.

**Solución:**

**Principio de penalización gradual**

En lugar de una penalización binaria (0 o 1), implementamos una penalización que crece monotónicamente con la distancia a la frontera original.

**Función de penalización lineal**

La función de pérdida relajada con penalización lineal es:

$$L_{\text{relajada}}(x, y; \boldsymbol{\theta}, \delta) = \begin{cases} 0 & \text{si } d(x, y) \leq 1 \\ \frac{d(x,y)-1}{\delta} & \text{si } 1 < d(x, y) \leq 1 + \delta \\ 1 & \text{si } d(x, y) > 1 + \delta \end{cases}$$

**Justificación matemática:**

**Región interior ($d \leq 1$):**

$$L_{\text{relajada}}(x, y) = 0$$

*Interpretación:* Clasificación completamente correcta, sin penalización.

**Zona de transición ($1 < d \leq 1 + \delta$):**

$$L_{\text{relajada}}(x, y) = \frac{d-1}{\delta} \in (0, 1)$$

*Interpretación:* Penalización proporcional a la distancia de la frontera.

**Región exterior ($d > 1 + \delta$):**

$$L_{\text{relajada}}(x, y) = 1$$

*Interpretación:* Penalización máxima para puntos claramente fuera de la hipótesis extendida.

**Propiedades de la función de penalización**

**Continuidad:** Verificamos que la función es continua en los puntos de transición:

*En $d = 1$:*

$$\lim_{d \to 1^-} L(d) = 0, \quad \lim_{d \to 1^+} L(d) = \frac{1-1}{\delta} = 0$$

*En $d = 1 + \delta$:*

$$\lim_{d \to (1+\delta)^-} L(d) = \frac{(1+\delta)-1}{\delta} = 1, \quad \lim_{d \to (1+\delta)^+} L(d) = 1$$

**Monotonicidad:** En la zona de transición:

$$\frac{dL}{dd} = \frac{1}{\delta} > 0$$

**Interpretación probabilística**

La función relajada puede interpretarse como la probabilidad de clasificación incorrecta:

$$L_{\text{relajada}}(x, y) \approx P(\text{clasificación incorrecta}|\text{posición en zona de transición})$$

donde la probabilidad aumenta linealmente con la distancia a la frontera.

d) Puedes elegir para este ejercicio una de las hipótesis de los ejercicios 1 y 2.

**Solución:**

**Elección: Hipótesis de elipse del Ejercicio 2**

**Justificación de la elección**

Elegimos la clase de hipótesis de elipses $\mathcal{H}_{\text{eli}}$ por las siguientes ventajas:

*a*) **Simplicidad computacional:** La distancia normalizada se calcula eficientemente

*b*) **Diferenciabilidad:** La función es diferenciable en casi todo punto

*c*) **Flexibilidad geométrica:** Las elipses pueden adaptarse a diferentes formas de datos

*d*) **Parametrización compacta:** Solo 4 parámetros $\{h, k, a, b\}$ más $\delta$

**Definición completa de la función relajada**

Para una elipse con parámetros $\boldsymbol{\theta} = \{h, k, a, b\}$ y parámetro de suavidad $\delta > 0$:

$$L_{\text{elipse}}(x, y; \boldsymbol{\theta}, \delta) = \begin{cases} 0 & \text{si } d(x, y) \leq 1 \\ \frac{d(x,y)-1}{\delta} & \text{si } 1 < d(x, y) \leq 1 + \delta \\ 1 & \text{si } d(x, y) > 1 + \delta \end{cases}$$

donde:

$$d(x, y) = \sqrt{\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2}}$$

**Familia de funciones relajadas**

Variando $\delta$, obtenemos una familia de funciones:

$$\mathcal{F}_{\text{relajada}} = \{L_{\text{elipse}}(\cdot; \boldsymbol{\theta}, \delta) : \boldsymbol{\theta} \in \mathbb{R}^4, \delta \in \mathbb{R}^+\}$$

**Casos límite:**

- $\delta \to 0^+$: Recuperamos la función indicadora binaria

- $\delta \to \infty$: La función se vuelve completamente suave pero pierde precisión

**Análisis de la superficie de pérdida**

La función $L_{\text{elipse}}(x, y)$ define una superficie en $\mathbb{R}^3$ con:

**Valores en regiones clave:**

- **Centro de la elipse:** $L(h, k) = 0$
- **Frontera original:** $L(x, y) = 0$ para $d(x, y) = 1$
- **Frontera extendida:** $L(x, y) = 1$ para $d(x, y) = 1 + \delta$

**Gradiente en la zona de transición:**

$$\nabla L(x, y) = \frac{1}{\delta} \cdot \frac{\nabla d(x, y)}{d(x, y)} \cdot \left( \frac{x - h}{a^2}, \frac{y - k}{b^2} \right)^T$$

e) Implementarás esto en la práctica asociada a este ejercicio.

**Solución:**

**Guía para la implementación**

La implementación debe incluir las siguientes componentes matemáticas:

**Función 1: Distancia normalizada**

$$d(\boldsymbol{x}; \boldsymbol{\theta}) = \sqrt{(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})}$$

donde $\boldsymbol{\mu} = (h, k)^T$ y $\boldsymbol{\Sigma} = \text{diag}(a^2, b^2)$.

**Función 2: Pérdida relajada**

$$L(\boldsymbol{x}; \boldsymbol{\theta}, \delta) = \max\left(0, \min\left(1, \frac{d(\boldsymbol{x}) - 1}{\delta}\right)\right)$$

**Función 3: Gradiente de la pérdida**

Para optimización, el gradiente respecto a los parámetros es:

$$\frac{\partial L}{\partial \boldsymbol{\theta}} = \begin{cases} \mathbf{0} & \text{si } d \leq 1 \text{ o } d > 1 + \delta \\ \frac{1}{\delta} \frac{\partial d}{\partial \boldsymbol{\theta}} & \text{si } 1 < d \leq 1 + \delta \end{cases}$$

**Algoritmo de optimización:**

**Algorithm 3** Optimización de Función de Pérdida Relajada

**Require:** $\{\boldsymbol{x}_i\}_{i=1}^n$, $\{y_i\}_{i=1}^n$, $\delta > 0$, $\alpha > 0$ (tasa de aprendizaje), $\epsilon > 0$ (tolerancia)
**Ensure:** $\boldsymbol{\theta}^*$ (parámetros óptimos)

▷ Inicialización
Inicializar $\boldsymbol{\theta}^{(0)}$ aleatoriamente
$t \leftarrow 0$
converged $\leftarrow$ false

**while** $\neg$converged **do**
  ▷ Calcular pérdida total
  $L_{\text{total}} \leftarrow \sum_{i=1}^n L(\boldsymbol{x}_i; \boldsymbol{\theta}^{(t)}, \delta)$

  ▷ Calcular gradientes
  $\nabla_{\boldsymbol{\theta}} L_{\text{total}} \leftarrow \sum_{i=1}^n \nabla_{\boldsymbol{\theta}} L(\boldsymbol{x}_i; \boldsymbol{\theta}^{(t)}, \delta)$

  ▷ Actualizar parámetros
  $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \alpha \nabla_{\boldsymbol{\theta}} L_{\text{total}}$

  ▷ Verificar convergencia
  **if** $\|\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\| < \epsilon$ **then**
    converged $\leftarrow$ true
  **end if**
  $t \leftarrow t + 1$
**end while**

**return** $\boldsymbol{\theta}^* = \boldsymbol{\theta}^{(t)}$

**Validación de la implementación:**

a) Verificar que $\lim_{\delta \to 0} L(\boldsymbol{x}) = L_{\text{hard}}(\boldsymbol{x})$
b) Comprobar continuidad en las fronteras
c) Validar monotonicidad en la zona de transición