

IA - Clase 2C

(ML – Machine Learning)

Aprendizaje de Máquina aplicado a imágenes – K-NN, K-means y GMM

Aprendizaje de Máquina (ML)

Problema de imágenes en ML

- Imagen digital = matriz de valores numéricos que representan la intensidad de luz y color.
- ML tratamos a la imagen como un tensor, para poder aplicar técnicas matemáticas y de optimización.
- Una imagen en blanco y negro puede representarse como:

$$\mathbf{X} \in \mathbb{R}^{H \times W}$$

- H: altura en pixeles
- W: ancho en pixeles
- Cada elemento $X(i,j) \in [0,255]$
- $[0,255]$ indica la intensidad (0 = negro, 255 = blanco)
 - Ejemplo: una imagen de 28×28 (MNIST) se representa como una matriz 28×28

Aprendizaje de Máquina (ML)

Problema de imágenes en ML

- Problemas típicos:
 - **Clasificación supervisada**: dado un conjunto de imágenes con etiquetas (perro, gato, auto), predecir la etiqueta de una nueva imagen.
 - **Clasificación no supervisada**: agrupar píxeles o imágenes sin etiquetas en categorías homogéneas.

Aprendizaje de Máquina (ML)

Algoritmo K-NN (K-Nearest Neighbors)

- Algoritmo supervisado de clasificación basado en la proximidad entre puntos.
- “Una imagen pertenece a la clase más común entre sus vecinos más cercanos”.
- Dado un dataset de entrenamiento:

$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

\mathbb{R}^p

- donde $x_i \in \mathbb{R}^p$ es el vector de características de la imagen e y_i es su etiqueta.
- Para clasificar una nueva imagen x :
 - Calcular la distancia entre x y todos los ejemplos x_i .
 - Seleccionar los K más cercanos.
 - Asignar la clase más frecuente donde $y_{(i)}$ es la etiqueta del i -ésimo vecino más cercano.

$$\hat{y}(x) = \arg \max_c \sum_{i=1}^K \mathbf{1}(y_{(i)} = c)$$

Aprendizaje de Máquina (ML)

Algoritmo K-Means (Clustering)

- Algoritmo no supervisado que agrupa puntos en K clusters minimizando la distancia a sus centroides.

- Objetivo:

$$J = \sum_{i=1}^N \sum_{k=1}^K r_{ik} \|x_i - \mu_k\|^2$$

- $r_{ik}=1$ si el punto x_i pertenece al cluster k , 0 en caso contrario.
- μ_k es el centroide del cluster k .

- Algoritmo

- Inicializar K centroides.
- Asignar cada punto al centroide más cercano.
- Recalcular centroides como la media de los puntos asignados.
- Repetir hasta convergencia.

- Aplicación a imágenes

- Segmentación de color: agrupar píxeles por similitud en espacio RGB o Lab → reducción de colores.
- Ejemplo: foto de paisaje segmentada en K=3 (cielo, montaña, agua).

Aprendizaje de Máquina (ML)

Algoritmo K-Means (Clustering)

■ Algoritmo

- Inicializar K centroides.
- Asignar cada punto al centroide más cercano.

$$r_{ik} = \begin{cases} 1 & \text{si } k = \arg \min_j \|x_i - \mu_j\|^2 \\ 0 & \text{caso contrario} \end{cases}$$

- Recalcular centroides como la media de los puntos asignados.

$$\mu_k = \frac{\sum_{i=1}^N r_{ik} x_i}{\sum_{i=1}^N r_{ik}}$$

- Repetir hasta convergencia.



Aprendizaje de Máquina (ML)

Algoritmo GMM (Gaussian Mixture Models)

- K-means asume clusters esféricos → limitación.
- K-means puede caer en mínimos locales → dependencia de inicialización.
- Algoritmo no supervisado, GMM modela clusters como combinación de distribuciones Gaussianas, clusters elípticos y más flexibles.

- Modelo:
$$p(x) = \sum_{k=1}^K \pi_k \mathcal{N}(x | \mu_k, \Sigma_k)$$
 - π_k = peso del cluster k.
 - μ_k, Σ_k = media y covarianza del cluster
 - $\sum_{k=1}^K \pi_k = 1$
 - $\mathcal{N}(x | \mu_k, \Sigma_k)$ = densidad normal multivariada.

Aprendizaje de Máquina (ML)

Algoritmo GMM (Gaussian Mixture Models)

- Algoritmo EM:
 - E-step: calcular probabilidad de pertenencia de x al cluster k .
 - M-step: actualizar parámetros.

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}(x_i | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(x_i | \mu_j, \Sigma_j)}$$

$$\mu_k = \frac{\sum_{i=1}^N \gamma_{ik} x_i}{\sum_{i=1}^N \gamma_{ik}}$$

$$\Sigma_k = \frac{\sum_{i=1}^N \gamma_{ik} (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^N \gamma_{ik}}$$

$$\pi_k = \frac{1}{N} \sum_{i=1}^N \gamma_{ik}$$

Aprendizaje de Máquina (ML)

Algoritmo GMM (Gaussian Mixture Models)

- Aplicación a imágenes
 - Segmentación probabilística: cada píxel puede pertenecer a varios clusters con distinta probabilidad.
 - Imágenes médicas (ej. separar tejido sano/enfermo).
 - Reconocimiento de texturas.

\hat{P}

Aprendizaje de Máquina (ML)

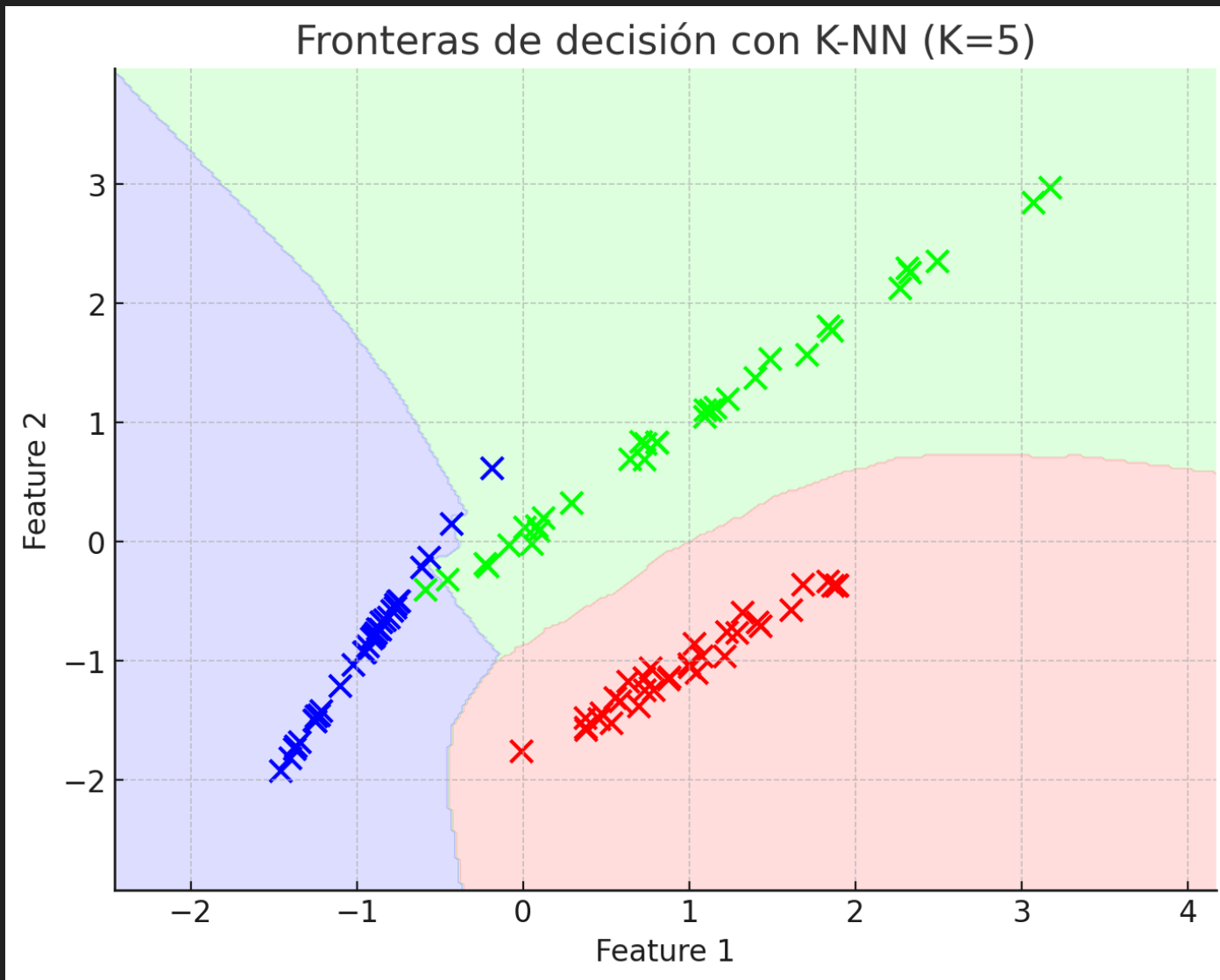
Clasificación de Imágenes

- **OBJETIVO:** asignar a cada imagen una etiqueta de clase de entre un conjunto finito de clases,
 - Distinguir entre dígitos escritos a mano (MNIST).
 - Reconocer perros vs. gatos.
 - Identificar objetos en ImageNet.
- **ENTRADA:** cada imagen se representa como un tensor $H \times W \times C$.
- **SALIDA:** variable aleatoria de clase donde K es el número de categorías posibles.
$$y_i \in \{1, 2, \dots, K\}$$
- **CLASIFICADOR:** se busca una función parametrizada donde θ son los parámetros (pesos del modelo).
$$f_{\theta} : \mathbb{R}^{H \times W \times C} \rightarrow \{1, \dots, K\}$$
- **ENTRENAMIENTO:** con pérdida. Se entrena sobre un conjunto de datos D minimizando la pérdida. Para casos de clasificación la pérdida es la entropía cruzada, donde \hat{y}^k es la probabilidad estimada de la clase k .

$$\mathcal{D} = \{(\mathbf{X}_i, y_i)\}_{i=1}^N \quad \mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N \ell(f_{\theta}(\mathbf{X}_i), y_i) \quad \ell(\hat{y}, y) = - \sum_{k=1}^K 1_{[y=k]} \log \hat{y}_k$$

Aprendizaje de Máquina (ML)

Clasificación de Imágenes



Aprendizaje de Máquina (ML)

Reinforcement Learning (RL)-Aprendizaje por Refuerzo

- Paradigma donde un agente aprende a tomar decisiones a través de la interacción con un entorno.
- En aprendizaje supervisado tenemos ejemplos de entrada-salida conocidos (etiquetados).
- En aprendizaje no supervisado buscamos una estructura oculta en los datos (clustering, reducción de dimensión, etc.).
- En aprendizaje por refuerzo, el agente no recibe la respuesta correcta explícita: aprende a partir de recompensas o castigos en función de sus acciones.

Aprendizaje de Máquina (ML)

Reinforcement Learning (RL)-Aprendizaje por Refuerzo

■ Componentes:

- **Agente**: el sistema que aprende (ej. un robot, un programa de trading, un personaje en un videojuego).
- **Entorno**: el “mundo” en el que actúa el agente (ej. el mercado, el videojuego, un simulador físico).
- **Estado (s)**: representa la situación actual del entorno (posición de un robot, tablero de ajedrez, etc.).
- **Acciones (a)**: las decisiones posibles del agente en un estado dado (mover arriba, comprar, girar, etc.).
- **Recompensa (r)**: señal escalar que indica lo bueno/malo de una acción en un estado (ej. ganar puntos, reducir pérdidas, alcanzar un objetivo).
- **Política (π)**: estrategia que define qué acción tomar en cada estado. Puede ser determinista o probabilística.
- **Función de valor (V o Q)**:
 - $V^\pi(s)$: valor esperado de un estado siguiendo la política π .
 - $Q^\pi(s,a)$: valor esperado de tomar acción a en estado s siguiendo π .



Aprendizaje de Máquina (ML)

Reinforcement Learning (RL)-Aprendizaje por Refuerzo

■ Objetivo:

- El objetivo del agente es maximizar la recompensa acumulada en el tiempo.
- Se suele usar la recompensa acumulada descontada donde $\gamma \in [0,1]$ es el factor de descuento (pondera más las recompensas inmediatas que las futuras).

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

- Ejemplo:

- Un robot en un laberinto:
- Estado: posición actual.
- Acciones: moverse arriba, abajo, izquierda, derecha.
- Recompensa: +1 si encuentra la salida, -1 si choca contra una pared.
- El robot aprende a seguir un camino que maximiza la recompensa esperada (llegar a la salida lo más rápido posible).

