Proyecto 2: Clustering

Jesús Valentín Niño Castañeda, Sebastian Hernandez Miñano, Persona 3, Persona 4,

***Resumen*—Este proyecto desarrolla un sistema de recomendación de películas mediante clustering no supervisado aplicado a características visuales de pósters cinematográficos. Se extrajeron descriptores de color, textura y forma de 9,337 pósters del dataset MovieLens, seguido de reducción de dimensionalidad con PCA, SVD y LDA.**

***Index Terms*—Clustering, Reducción de dimensionalidad, Computer Vision, Sistemas de recomendación, Movie posters, Aprendizaje no supervisado**

I. Introducción

Los sistemas de recomendación son herramientas fundamentales en la era digital, permitiendo a los usuarios descubrir contenido relevante dentro de vastas bibliotecas multimedia. En el contexto cinematográfico, los métodos tradicionales se basan principalmente en información textual como géneros, directores, actores y reseñas de usuarios. Sin embargo, los pósters de películas constituyen una fuente rica de información visual que comunica elementos estéticos, temáticos y emocionales, características que influyen significativamente en las preferencias del espectador. Este proyecto explora un enfoque alternativo basado en técnicas de computer vision y clustering no supervisado para analizar y agrupar películas según la similitud visual de sus pósters, ofreciendo recomendaciones fundamentadas en características perceptuales en lugar de metadatos convencionales.

I-A. Objetivos

El objetivo principal de este proyecto es desarrollar un sistema de recomendación de películas basado en el análisis visual de pósters cinematográficos mediante técnicas de agrupamiento no supervisado. Para ello, se extraerán características visuales utilizando métodos tradicionales de computer vision, se aplicarán técnicas de reducción de dimensionalidad para optimizar la representación de los datos, y se implementarán diversos algoritmos de clustering para identificar grupos de películas con similitud visual. Finalmente, se evaluará la calidad de las agrupaciones mediante métricas internas y análisis de coherencia con géneros cinematográficos, permitiendo validar la efectividad del enfoque visual para la recomendación de contenido.

Este trabajo fue realizado como parte del curso de ML en la UTEC.

II. Dataset

II-A. Descarga y Preparación de Pósters

La fase inicial del preprocesamiento consistió en la descarga automatizada de las imágenes de pósters desde las URLs proporcionadas en el dataset. Se implementó un sistema robusto de descarga con manejo de excepciones, timeouts configurables y control de tasa de peticiones para respetar las políticas de uso de la API de TMDb. Del total de 9,337 pósters, se logró descargar exitosamente 9,334 imágenes en formato JPEG, representando una tasa de éxito del 99.97 %. Las tres descargas fallidas fueron reintentadas sin éxito debido a URLs inválidas o contenido removido de la plataforma.

Todas las imágenes descargadas se almacenaron localmente con nomenclatura estandarizada utilizando el *movieId* como identificador único (e.g., `619.jpg`), facilitando el acceso eficiente durante las etapas posteriores de extracción de características. No se aplicó redimensionamiento uniforme en esta fase, preservando las dimensiones originales de cada póster para mantener la integridad de la información visual.

II-B. Extracción de Características Visuales

Para representar las características visuales de cada póster cinematográfico, se implementaron cuatro familias complementarias de descriptores que capturan diferentes aspectos de la información visual: color, textura, forma y geometría global.

II-B1. Histogramas de Color (HSV): Se extrajeron histogramas tridimensionales en el espacio de color HSV (Hue-Saturation-Value), preferido sobre RGB por su mayor robustez ante variaciones de iluminación y su alineación con la percepción humana del color. Para cada canal se calcularon histogramas

con 64 bins: canal H en el rango [0, 180°], canales S y V en [0, 256]. Los tres histogramas se concatenaron y normalizaron mediante la suma total más un término de regularización ($\epsilon = 10^{-7}$) para evitar divisiones por cero, resultando en vectores de 192 dimensiones que capturan la distribución cromática global del póster.

II-B2. Descriptores de Textura (LBP): Se utilizaron Local Binary Patterns (LBP) con configuración uniforme para caracterizar la micro-textura de cada imagen. Específicamente, se aplicó LBP con 64 puntos de muestreo y radio 3 sobre la versión en escala de grises de cada póster. El descriptor uniforme reduce significativamente la dimensionalidad al considerar solo patrones con a lo más dos transiciones 0-1 en el código binario circular, resultando en histogramas de 66 bins. Esta representación es invariante a transformaciones monotónicas de intensidad y efectiva para distinguir entre estilos visuales como fotografía realista versus diseño gráfico ilustrativo.

II-B3. Descriptores de Forma y Bordes (HOG): Para capturar la estructura espacial y composición de cada póster, se extrajeron características Histogram of Oriented Gradients (HOG). Cada imagen se redimensionó a 64×96 píxeles y se convirtió a escala de grises. Se configuró HOG con 10 orientaciones, celdas de 8×8 píxeles y bloques de normalización de 2×2 celdas, produciendo vectores de 3,780 dimensiones. Estos descriptores son particularmente efectivos para capturar la disposición de elementos visuales dominantes como siluetas de personajes, texto tipográfico y composición geométrica.

II-B4. Momentos Geométricos (Hu Moments): Finalmente, se calcularon los siete momentos invariantes de Hu a partir de los momentos centrales normalizados de cada imagen en escala de grises. Estos descriptores son invariantes a traslación, rotación y escala, capturando propiedades geométricas globales como simetría, balance y estructura espacial. Para mejorar la estabilidad numérica, se aplicó la transformación logarítmica: $-\text{sign}(h_i) \cdot \log_{10}(|h_i| + 10^{-10})$, donde h_i representa cada momento de Hu.

II-B5. Vector de Características Combinado: La concatenación de los cuatro descriptores produce vectores de características finales de 3,345 dimensiones (192 + 66 + 3,780 + 7) en formato `float32` para cada póster. Esta representación multimodal captura aspectos complementarios de la información visual: el histograma HSV codifica el contenido cromático, LBP la textura superficial, HOG la estructura compositiva, y los momentos de Hu las propiedades geométricas globales. Todos los vectores se almacenaron en formato comprimido NPZ junto con sus correspondientes *movieIds* para facilitar el procesamiento posterior.

II-C. Reducción de Dimensionalidad

Los vectores de características de 3,345 dimensiones presentan desafíos computacionales y riesgos de sobreajuste en algoritmos de clustering. Para mitigar estos problemas, se aplicaron tres técnicas de reducción de dimensionalidad, cada una con propiedades matemáticas distintas, reduciendo la representación a 64 componentes principales.

II-C1. Estandarización Previa: Previo a la aplicación de cualquier técnica de reducción, todos los vectores de características se estandarizaron utilizando `StandardScaler` de scikit-learn, transformando cada dimensión para tener media cero y varianza unitaria. Este paso es crítico para garantizar que características con escalas naturalmente mayores (como los 3,780 componentes de HOG) no dominen el análisis sobre características más compactas (como los 7 momentos de Hu).

II-C2. Análisis de Componentes Principales (PCA): PCA realiza una descomposición de valores propios de la matriz de covarianza de los datos, identificando direcciones ortogonales de máxima varianza. Se configuró para retener 64 componentes principales, completando el ajuste en 1.53 segundos. La varianza explicada acumulada por estos 64 componentes fue de 35.76 %, indicando que aproximadamente un tercio de la variabilidad total del espacio original se preserva en la proyección reducida. Esta técnica es puramente no supervisada y no utiliza información de etiquetas de género.

II-C3. Descomposición en Valores Singulares (SVD): Truncated SVD, matemáticamente equivalente a PCA pero implementada mediante descomposición de valores singulares, se aplicó directamente sobre la matriz de características estandarizadas. Esta técnica completó el procesamiento en 1.18 segundos (23 % más rápida que PCA) y capturó 35.70 % de la varianza explicada con 64 componentes. La ligera diferencia en varianza explicada respecto a PCA (0.06 %) se debe a diferencias numéricas en la implementación algorítmica, siendo prácticamente equivalentes en términos de información preservada.

II-C4. Análisis Discriminante Lineal (LDA): A diferencia de PCA y SVD, LDA es una técnica supervisada que maximiza la separabilidad entre clases. Para su aplicación, se utilizaron las etiquetas de género primario (el primer género listado en cada película) como variable objetivo. Se filtraron géneros con menos de 20 muestras para garantizar estabilidad estadística, resultando en 16 clases distintas. El número de componentes discriminantes está limitado por $\min(n_{clases}-1, n_{features})$, resultando en 15 componentes para LDA. El proceso requirió 8.23 segundos y alcanzó una varianza explicada de 100 % entre las 15 dimensiones discriminantes, utilizando 9,321 de las 9,337 películas originales (99.83 %).

III. Parte Teórica (30 pts)

III-A. Fundamentos de K-means y GMM. (10 pts)

Considere algoritmos de agrupamiento para datos

$$D = \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\} \text{ con } x^{(i)} \in \mathbb{R}^d,$$

- La función objetivo de K-means $J(\mu_1, \ldots, \mu_k, z) = \sum_{i=1}^{N} \sum_{j=1}^{k} z_{ij} ||x^{(i)} - \mu_j||^2$ nunca aumenta durante las iteraciones debido a la naturaleza de los dos pasos del algoritmo, donde cada uno minimiza J respecto a sus variables correspondientes. En el paso de **Expectation**, manteniendo fijos los centroides μ_j, cada punto $x^{(i)}$ se asigna al cluster cuyo centroide está más cercano mediante $z_{ij} = 1$ si $j = \arg\min_k ||x^{(i)} - \mu_k||^2$. Esta asignación minimiza la contribución de cada punto a la función objetivo, ya que por definición se está eligiendo el centroide que produce la menor distancia cuadrática. Por lo tanto, J después de este paso es menor o igual que antes. En el paso de **Maximization**, con las asignaciones z_{ij} fijas, los centroides se actualizan como $\mu_j = (\sum_i z_{ij} x^{(i)})/(\sum_i z_{ij})$, que corresponde a la media aritmética de los puntos asignados a cada cluster. Esta actualización minimiza la suma de distancias cuadráticas dentro de cada cluster, lo cual puede demostrarse tomando la derivada de J respecto a μ_j e igualando a cero. Nuevamente, J después de este paso es menor o igual que antes. Como ambos pasos garantizan que J no aumente y la función está acotada inferiormente por 0, el algoritmo converge cuando los centroides dejan de cambiar significativamente. Sin embargo, esta convergencia no garantiza el óptimo global porque K-means es un algoritmo voraz que realiza optimización local en cada iteración. La función objetivo es no convexa respecto a (μ, z) conjuntamente, presentando múltiples mínimos locales. El algoritmo converge al mínimo local más cercano a la inicialización aleatoria de los centroides, por lo que diferentes inicializaciones pueden producir soluciones finales distintas con valores diferentes de J.
- **K-means** asume que los clusters tienen forma esférica con igual varianza en todas las direcciones, equivalente a modelar cada cluster como una distribución gaussiana isotrópica con matriz de covarianza $\sigma^2 I$. Todos los clusters tienen aproximadamente el mismo tamaño y la asignación es determinística (hard clustering), donde cada punto pertenece exactamente a un cluster mediante la minimización de la distancia euclidiana.

  GMM (Gaussian Mixture Models) modela los datos como una mezcla de distribuciones gaussianas $p(x) = \sum_j \pi_j \mathcal{N}(x|\mu_j, \Sigma_j)$, permitiendo clusters elípticos con diferentes orientaciones mediante matrices de covarianza completas Σ_j. Cada cluster puede tener diferente tamaño, forma y densidad según sus parámetros π_j (peso), μ_j (media) y Σ_j (covarianza). La asignación es probabilística (soft clustering), calculando la probabilidad posterior de pertenencia a cada cluster.

  Ejemplo donde GMM supera a K-means: Considere dos clusters elongados en $\mathbb{R}^2$ con orientaciones diferentes: el Cluster 1 contiene puntos distribuidos a lo largo de una elipse diagonal estrecha centrada en $(2, 2)$ con alta varianza en la dirección $(1, 1)$, mientras que el Cluster 2 tiene puntos distribuidos en una elipse anti-diagonal centrada en $(6, 2)$ con alta varianza en la dirección $(1, -1)$.

  En este escenario, K-means fallaría al intentar ajustar círculos a datos claramente elípticos, mezclando puntos de ambos clusters en las regiones donde se superponen las esferas. GMM, en cambio, puede estimar correctamente las matrices de covarianza que capturan la forma elongada y orientación diagonal de cada cluster, modelando apropiadamente la estructura

real de los datos mediante gaussianas con matrices de covarianza no isotrópicas que se adaptan a la distribución observada.

III-B. Agrupamiento Jerárquico. (10 pts)

Para un conjunto de datos

$$D = \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\} \text{ con } x^{(i)} \in \mathbb{R}^d,$$

Para dos clusters C_i y C_j, las métricas de distancia se definen matemáticamente como:

Enlace Simple: $d_{\text{single}}(C_i, C_j) = \min\{\|x - y\| : x \in C_i, y \in C_j\}$

Corresponde a la distancia mínima entre cualquier par de puntos de los dos clusters. Tiende a formar clusters elongados y es sensible a puntos puente.

Enlace Completo: $d_{\text{complete}}(C_i, C_j) = \max\{\|x - y\| : x \in C_i, y \in C_j\}$

Representa la distancia máxima entre cualquier par de puntos de los dos clusters. Produce clusters compactos y esféricos, siendo sensible a outliers.

Enlace Promedio: $d_{\text{average}}(C_i, C_j) = \frac{1}{|C_i| \times |C_j|} \sum_{x \in C_i} \sum_{y \in C_j} \|x - y\|$

Calcula el promedio de todas las distancias entre pares de puntos de ambos clusters. Ofrece un balance entre los enfoques simple y completo.

Dados los puntos: $P_1 = (1, 1), P_2 = (2, 1), P_3 = (5, 3), P_4 = (6, 3), P_5 = (10, 5)$ **Con enlace Simple:** *Primera fusión:* Los pares más cercanos son (P_1, P_2) y (P_3, P_4) con distancia 1,0. Fusionamos P_1 y P_2 formando $C_1 = \{P_1, P_2\}$.

Segunda fusión: Fusionamos P_3 y P_4 formando $C_2 = \{P_3, P_4\}$ con distancia 1,0.

Las distancias se recalculan como:

- $d_{\text{single}}(C_1, C_2) = \min\{d(P_1, P_3), d(P_1, P_4), d(P_2, P_3), d(P_2, P_4)\} =$ 3,61
- $d_{\text{single}}(C_1, P_5) = \min\{d(P_1, P_5), d(P_2, P_5)\} =$ 8,06
- $d_{\text{single}}(C_2, P_5) = \min\{d(P_3, P_5), d(P_4, P_5)\} =$ 4,47

Con Enlace Completo: Las primeras fusiones serían las mismas (P_1, P_2) y (P_3, P_4), pero las distancias entre clusters se calculan diferentemente: $d_{\text{complete}}(C_1, C_2) = \max\{d(P_1, P_3), d(P_1, P_4), d(P_2, P_3), d(P_2, P_4)\} =$ 5,39 El enlace completo produce clusters más compactos y podría fusionar C_2 con P_5 antes que C_1 con C_2, resultando en una estructura jerárquica diferente donde se favorecen agrupamientos con menor diámetro máximo.

III-C. Agrupamiento Basado en Densidad (10 pts)

- **Puntos directamente alcanzables por densidad:** Un punto q es directamente alcanzable por densidad desde un punto p si:
  - $\|p - q\| \leq \epsilon$ (q está en el ϵ-vecindario de p)
  - $|N_\epsilon(p)| \geq$ MinPts (p es un punto núcleo)

  donde $N_\epsilon(p) = \{q \in D : \|p - q\| \leq \epsilon\}$ es el ϵ-vecindario de p.
- **Puntos alcanzables por densidad:** Un punto q es alcanzable por densidad desde p si existe una cadena de puntos $p_1, \ldots, p_n$ donde $p_1 = p, p_n = q$, y cada p_{i+1} es directamente alcanzable por densidad desde p_i. Es la clausura transitiva de la alcanzabilidad directa.
- **Puntos conectados por densidad:** Dos puntos p y q están conectados por densidad si existe un punto o tal que tanto p como q son alcanzables por densidad desde o. Esto define una relación simétrica que forma la base para la creación de clusters.

Para ilustrar cómo DBSCAN puede identificar clusters de forma arbitraria mientras K-means falla, voy a usar un ejemplo con datos en forma de dos medias lunas entrelazadas.

Descripción del conjunto de datos:

Imaginemos dos medias lunas en el plano 2D que se entrelazan pero no se tocan. La primera media luna tiene forma de C abierta hacia la derecha, mientras que la segunda está rotada y posicionada de manera que se entrelaza con la primera sin que los puntos se mezclen. Este tipo de datos es común en problemas reales donde los clusters no tienen formas esféricas simples.

¿Cómo funciona DBSCAN en este caso?

Usando parámetros adecuados como $\epsilon = 0{,}3$ (radio de vecindario) y MinPts $= 5$ (puntos mínimos para formar un cluster), DBSCAN va a:

- Primero identificar los puntos núcleo, que son aquellos que tienen al menos 5 puntos en su vecindario de radio 0,3
- Luego expandir desde estos puntos núcleo siguiendo la densidad de puntos
- Como los puntos dentro de cada media luna están cerca entre sí, DBSCAN los va conectando progresivamente
- Al final, obtiene dos clusters que respetan perfectamente la forma de media luna original

¿Por qué K-means no funciona bien aquí?

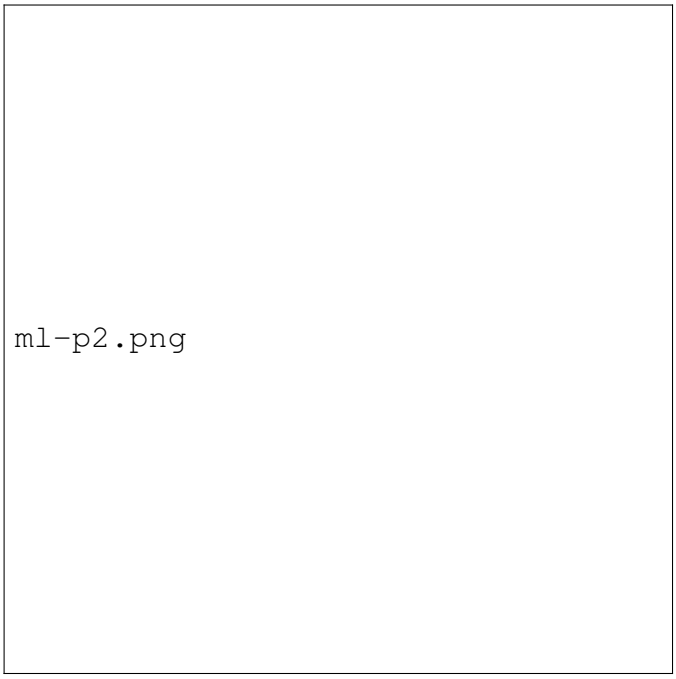

Figura 1. Comparación entre K-means y DBSCAN

Si aplicamos K-means con $k = 2$ al mismo conjunto de datos:

- El algoritmo va a buscar dos centroides que minimicen la suma de distancias cuadráticas
- Como K-means asume clusters esféricos, va a intentar dividir el espacio en dos regiones convexas
- Esto resulta en una división que probablemente corte ambas medias lunas por la mitad
- Cada cluster resultante va a contener pedazos de ambas medias lunas, mezclando las estructuras originales

La razón principal de esta diferencia es que DBSCAN se basa en la densidad local y puede seguir formas curvas conectando puntos cercanos, mientras que K-means divide el espacio según la distancia a centroides fijos, creando fronteras lineales que no pueden adaptarse a formas no convexas.

IV. Parte Aplicada (70 pts)

IV-A. Selección y Preprocesamiento del Dataset

IV-B. Metodología de Modelado

IV-C. Experimentación y Resultados

IV-D. Discusión

V. Conclusiones

VI. Contribution Statement

- **Jesús Valentín Niño Castañeda (100 %):** Preprocesamiento del dataset (unión de ids), descargar de posters, extracción de características, reducción de dimensionalidad, desarrollo de modelos de clustering .
- **Sebastian Hernandez Miñano (100 %):** Resolución de la parte teórica y preparación del Dataset .
- **Persona 3**
- https://github.com/Jvnc0503/ML-Project2:

Referencias

[1] Zhang, L., & Wang, L. (1995). A survey of content-based image retrieval. *Journal of Visual Communication and Image Representation*, 6(2), 135-149.