Proyecto Final: Sistema de Clasificación de datos Multimedia

Universidad Carlos III de Madrid, Multimedia, Curso 2022-2023

Introducción y objetivos

El procesado de datos multimedia es un campo de trabajo muy versátil, con multitud de aplicaciones en distintos campos de la ingeniería. Una de esas aplicaciones es la resolución de problemas de clasificación.

En este proyecto se desarrollará un sistema de reconocimiento de géneros de películas. Para ello, nos centraremos en las características visuales de los carteles de las películas, así como en las características de texto de su descripción, sinopsis, reparto, etc. Los datos de trabajo han sido extraídos de la popular plataforma de cine IMDb[1].

Figura 1. Ejemplos de carteles de películas

Los objetivos concretos se pueden resumir en los siguientes puntos:

- La extracción efectiva de características visuales y textuales que representen de manera simple el objeto bajo análisis.
- La clasificación de las películas en los géneros cinematográficos de comedia y drama, basándose en las características extraídas.

Al finalizar este proyecto, se manejarán con soltura algunas de las principales herramientas básicas de procesado de imagen y texto.

[1] https://www.imdb.com/

Entrega de la memoria ⚠

Importante: para la evaluación de este ejercicio, es necesario subir a Aula Global un archivo comprimido (en formato zip) que contenga lo siguiente:

- Un documento ***en formato .pdf*** siguiendo el formato que se establece en el archivo ***plantilla_proyecto***, disponible en Aula Global. El documento entregado en el archivo comprimido en formato zip tendrá un **máximo de 10 páginas** y debe contener las respuestas a las preguntas formuladas en las actividades presentes en el apartado Desarrollo. **Se recomienda muy encarecidamente proporcionar figuras e imágenes que ilustren los argumentos propuestos en las respuestas.**
- Un directorio que contenga todos los archivos de MATLAB con los que se ha trabajado. Este directorio debe funcionar de manera independiente al importarse como proyecto a MATLAB.

  Nota: Es suficiente con que un único miembro del grupo cuelgue el proyecto en Aula Global.

Desarrollo

El desarrollo de este proyecto se dividirá en 4 fases principales:

1. Extracción de características visuales
2. Extracción de características de texto
3. Entrenamiento y clasificación
4. Evaluación del rendimiento del sistema

Para algunas de estas fases será necesario implementar porciones de código, mientras que para otras se requerirá ejecutar código ya implementado y reflexionar sobre los resultados producidos.

En este punto, es necesario descargar el material del proyecto, disponible en Aula Global. Este material se agrupa en 3 elementos principales:

- Un script de Matlab (***skeleton.m***), en el que se implementarán las soluciones a los problemas planteados a lo largo del desarrollo. Este script está diseñado para ser completado, por lo que si se ejecuta sin ser modificado no funcionará correctamente.
- Una librería de funciones útiles (***lib***) que serán utilizadas durante el desarrollo.
- Una base de datos (***data***) que contiene todas las imágenes y el texto necesarios para el desarrollo del sistema.

A continuación, se procederá al desarrollo del sistema a través de sus respectivas fases. Antes de comenzar con la primera fase del proyecto, nos familiarizaremos con la base de datos que se utilizará a lo largo del mismo.

Fase 0: Generación de base de datos

El primer paso en cualquier proyecto de desarrollo de sistemas de clasificación es la generación de una base de datos sobre la que trabajar. En este caso, la base de datos se proporciona como parte del problema, por lo que este paso será mucho más sencillo.

Abra el script skeleton.m en Matlab y deténgase en la primera sección de código. Pruebe a ejecutar únicamente dicha sección.

> **Nota**: Recuerde que puede ejecutar una única sección del código haciendo click en cualquier línea de la misma y usando el comando Ctrl+Enter. También puede seleccionar partes del código y comentarlas a través del uso del comando Ctrl+R (puede descomentarlas usando Ctrl+T).

Esta acción debería desencadenar la creación de 6 variables distintas en su espacio de trabajo (workspace), que corresponden a los datos que usaremos para implementar el sistema de discriminación de géneros cinematográficos. En concreto, tenemos dos conjuntos de datos: uno de entrenamiento (train) y otro de evaluación (test). Para cada conjunto, contamos con una variable X, que contiene las muestras (carteles de películas y textos descriptivos) y una variable Y, que contiene una etiqueta (un número entre cero -comedia- y uno -drama- que actúa de identificador) para cada muestra.

> **Nota**: Puede interactuar con estas variables y observar sus contenidos haciendo doble click sobre ellas en el workspace.

Trate de familiarizarse con cada variable y acceder a los campos que la forman antes de continuar.

Fase 1: Extracción de características visuales

El objetivo principal de la extracción de características es el de representar de manera sencilla datos más complejos. En este caso, contamos con carteles de películas representados en imágenes en color (3 canales RGB) de 268x182 píxeles por imagen. Es decir, que incluso en imágenes de baja resolución como las que se manejan aquí, se necesitan miles de valores para representar una única muestra.

A través del uso de técnicas de procesado de imagen, en esta fase se pretende representar cada muestra únicamente con 3 valores. La dificultad de este proceso reside pues en seleccionar adecuadamente cómo extraer estos valores para que resulten decisivos a la hora de discernir entre distintos tipos de carteles de cine. Con esto en mente, se va a proceder a la extracción de (1) la variabilidad del color, (2) la luminancia y (3) la cantidad de información de bordes de la imagen analizada.

> **P1**. Reflexione sobre lo explicado en clase sobre esta etapa del proceso. ¿Por qué extraemos características en vez de alimentar la imagen entera al clasificador?

Variabilidad del color

Para extraer una característica del color, es útil hacer uso de la componente de matiz del espacio de color HSV y, en concreto, de uno de sus canales: el matiz. En este caso, tratamos de averiguar cuánto varía dicha componente de color en la imagen bajo análisis. Haremos uso del concepto de entropía. Complete el código para extraer la componente de color y calcule la característica (feature) como la entropía de los valores de dicha componente en el objeto bajo análisis. Haga uso de las funciones **`rgb2hsv`** y **`entropy`** de Matlab.

> **P2**. ¿Qué ventajas ofrece el espacio de color HSV sobre el RGB?

Luminancia

La intensidad de luz de una imagen nos da mucha información acerca del contenido de la misma. Por ejemplo, en fotos realizadas en exteriores, nos puede dar una idea de la hora aproximada del día en la que se han capturado. Aquí, se hará uso de la función **`mean`** para extraer la característica de luminancia como la intensidad media del canal de valor (HS**V**) de la imagen.

> **P3**. Se han utilizado los canales de Matiz (H) y Valor (V), ¿qué representa el canal restante? Describa en sus propias palabras la utilidad de dicho canal.

Cantidad de bordes

Hacer un análisis del gradiente de una imagen nos puede servir, entre otras cosas, para extraer la información del contorno de los elementos que contiene dicha imagen. Haga uso de la función **edge** para extraer, mediante el método de Sobel, los bordes de los carteles. A continuación, extraiga la tercera y última característica visual como la cantidad total de píxeles que se consideran pertenecientes a un borde.

> **P4**. En este caso, todas las imágenes tienen la misma resolución. Si no hubiera sido así, ¿sería útil esta característica tal y como la hemos extraído? Explique por qué y, en caso negativo, explique cómo podría solucionarse este problema.

En este punto, se cuenta con 3 valores que representan de manera concisa cada muestra (en este caso, un cartel de película en color de resolución 268x182).

Los datos de entrada de nuestro sistema de descripción visual son imágenes de carteles cinematográficos. Para las características visuales seleccionadas, no es necesario realizar ningún preprocesado sobre las imágenes. Sin embargo, a menudo se utilizan técnicas como la segmentación o la morfología matemática con el objetivo de preparar nuestras imágenes para una extracción de características más eficiente.

> **P5**. ¿Qué utilidad podría tener la aplicación de técnicas de segmentación (en concreto, el empleo del **método de Otsu**) sobre las imágenes de carteles de películas?

Fase 2: Extracción de características de texto

Una de las ventajas de los sistemas de clasificación basados en descriptores es que una vez se extraen las características, éstas pasan a ser simples valores que representan a una muestra determinada. De esta forma, se pueden mezclar distintos tipos de información

(siempre que representen al mismo elemento) para alimentar un mismo sistema de clasificación.

En este caso, además de contar con los carteles de películas, disponemos de un archivo de texto con contenido descriptivo de cada una. Este archivo ha sido extraído directamente de IMDb y, en concreto, del archivo HTML (sin etiquetas) que contiene la página de cada película. Así, se trata de un archivo bastante crudo que precisará de bastante procesado para resultar útil.

Este tipo de procesado se engloba en un campo de trabajo denominado Procesado del Lenguaje Natural (PLN) y se introducirá con detalle en la segunda parte de la asignatura. No obstante, a continuación veremos cómo extraer dos sencillas características basadas en este documento de texto.

> **P6**. Dada la configuración actual del sistema, ¿cuántos elementos tendrá en total nuestra matriz de características de entrenamiento una vez extraídas estas dos características para todas las muestras? Justifique su respuesta.

Para este caso introductorio, se realiza un procesado del lenguaje sencillo para separar el archivo en palabras individuales a través del empleo de la función **`obtain_word_array`**. Para este proyecto, por tanto, puede considerar la variable resultante (**words**) como un vector de las palabras que contiene la descripción de cada película.

> **P7**. Tomando como referencia la película Made of Honor (de la que se muestra el cartel en la Fig. 2), se proporciona la siguiente lista de palabras extraídas aleatoriamente del vector **words** correspondiente:
>
> - the
> - to
> - maid
> - Romance
> - of
> - wedding
>
> Responda las siguientes 3 preguntas: ¿Qué palabras se repetirán más en dicho vector **words**?¿Cuáles se repetirán más a lo largo de todos los documentos?¿Cuáles resultarán más útiles a la hora de distinguir esta película por su género? Justifique sus respuestas.

Figura 2. Cartel de la película Made of Honor (2008).

Complete el código para seleccionar el documento de texto correspondiente de los datos de entrenamiento y extraer las características de texto como la cantidad de palabras en el archivo de texto descriptivo y la longitud media de dichas palabras. Puede hacer uso de las funciones **`length`** y **`strlength`** de Matlab.

Fase 3: Entrenamiento y clasificación

Si se ha realizado correctamente la extracción de características, en este momento contaremos con una variable llamada **`features`** en el workspace, que contará con 960 filas, correspondientes a cada muestra de entrenamiento, y 5 columnas, correspondientes a cada una de las características que se han extraído durante las fases anteriores. Sin embargo, si analizamos dicha variable, podemos observar que el rango de valores varía enormemente de una característica a otra. Esto perjudica al sistema, ya que puede dar más peso a unas características que a otras a la hora de realizar la clasificación. Por este motivo, es importante normalizar esta variable antes de entrenar nuestro sistema de clasificación.

Implemente la normalización de la variable features por columnas, es decir, normalice por separado cada característica. Recuerde que la fórmula para la normalización de un vector x es la siguiente:

$$x_N = \frac{x - \mu_x}{\sigma_x}$$

siendo μ_xy σ_x la media y desviación típica del vector x, respectivamente. Haga uso de las funciones **`mean`** y **`std`** de Matlab.

Puede comprobar si ha realizado correctamente la normalización haciendo uso de la función **`check_normalization`**.

Antes de continuar con la siguiente fase del proyecto, detengámonos a reflexionar sobre la utilidad de las características extraídas. En el esqueleto puede encontrarse una sección dedicada a la visualización de dichas características (Feature Visualization). A través de dicho código, puede elegir representar (de dos en dos) los valores de las características para todas las muestras mediante un diagrama de dispersión. Pruebe a representar las distintas combinaciones de las 3 primeras características (características visuales). Basándose en dichos diagramas de dispersión, responda a las siguientes preguntas.

> **P8**. ¿Qué característica considera más útil para la discriminación de géneros cinematográficos? Justifique su respuesta y **apóyese en figuras**.
>
> **P9.** ¿Qué característica considera menos relevante para la discriminación de géneros cinematográficos? Justifique su respuesta y **apóyese en figuras**.
>
> **P10.** Escoja aleatoriamente dos características y represente su diagrama de dispersión. ¿Considera que las dos características escogidas son efectivas para discriminar géneros cinematográficos? Justifique su respuesta y **apóyese en figuras**.

P11. Teniendo en cuenta el desarrollo actual y los conocimientos aprendidos, mencione al menos 2 características adicionales (ya sean visuales o de texto) que se podrían extraer para este sistema. Justifique su elección.

***Ejemplo**: Calcular la mediana del canal de saturación (H**S**V).*

Una vez contamos con la matriz de características normalizada, podemos proceder al entrenamiento de nuestro sistema. El objetivo es, mediante el valor de las características extraídas y el conocimiento que tenemos sobre su contenido (las etiquetas que determinan su clase), obtener un modelo que sea capaz de clasificar futuras muestras entre las 2 clases empleadas para el entrenamiento.

Para el entrenamiento de este sistema, se empleará un clasificador Gaussiano sencillo. Al tratarse de un problema de clasificación binario (sólo 2 clases), se considerará una de las clases como clase positiva (clase a detectar) y la otra como clase negativa (clase a evitar). Esta nomenclatura es muy útil cuando se manejan clases claramente excluyentes. Por ejemplo, a la hora de analizar imágenes médicas: cáncer (clase positiva), no cáncer (clase negativa). Sin embargo, para este caso la elección de la clase positiva entre las dos disponibles es meramente anecdótica. Abra la función **`fit_gaussian`**, localizada en el directorio lib, y trate de comprender el código que se presenta en ella.

A continuación, debe analizar una película del conjunto de entrenamiento de la base de datos. Para seleccionarla, debe sumar todos los dígitos de los NIA de todos los integrantes del grupo. El número resultante será el índice de la película cuyo cartel debe analizar (haga constancia de este cálculo en la memoria). Se recomienda el desarrollo de un nuevo script (no es necesario entregarlo, aunque sí recomendable) para implementar el código necesario para responder a las próximas 3 preguntas.

P12. Según el modelo desarrollado, ¿a qué clase pertenece la película bajo análisis? Justifique su respuesta.

P13. ¿Cómo de seguro está el modelo de la pertenencia a la clase seleccionada? Justifique su respuesta.

P14. ¿Ha acertado en la clasificación? En caso positivo, ¿qué característica cree que ha sido más útil en la discriminación? En caso negativo, ¿por qué cree que ha fallado el sistema? Es recomendable consultar los valores de las características, así como volver a consultar los diagramas de dispersión para contestar a esta pregunta.

Una vez entrenado, nuestro sistema de clasificación está completo. Como puede observar, entrenamos tres modelos de clasificación distintos: (1) utilizando todas las características extraídas, (2) utilizando únicamente las características visuales y (3) utilizando exclusivamente las de texto. No obstante, esto no sirve de mucho si no podemos proporcionar alguna cifra sobre su fiabilidad. Para ello, debemos observar cómo se comportan con muestras que no hayan visto nunca con anterioridad.

Fase 4: Evaluación del rendimiento del sistema

Recordemos que, al principio del desarrollo del proyecto, separamos nuestra base de datos de imágenes en dos conjuntos: el de entrenamiento, empleado durante la fases anteriores

para obtener nuestro sistema de clasificación, y el de evaluación, que emplearemos en esta fase para evaluar nuestro sistema. Para ello, vamos a evaluar tres versiones distintas del mismo: (1) utilizando únicamente las características visuales, (2) utilizando únicamente las características textuales y (3) utilizando ambos conjuntos de características para obtener el modelo global.

Las muestras de evaluación tienen que someterse al mismo procesado que las de entrenamiento para poder extraer sus características. Complete el código de extracción de características y normalización para las nuevas imágenes.

> **Nota**: Puede reutilizar código de las secciones anteriores.

Recuerde que, para la normalización, no deben recalcularse los valores de media y desviación típica, sino que han de emplearse los valores obtenidos durante el entrenamiento del sistema.

En este caso, contamos con un conjunto de evaluación que supone el 40% de los datos totales (el 60% restante se dedica a entrenamiento).

> **P15**. ¿Por qué utilizamos dos conjuntos distintos para la fase de entrenamiento y para la fase de evaluación?
>
> **P16**. Teniendo en cuenta que el tamaño total de la base de datos no varía, ¿qué riesgo corremos al utilizar un porcentaje muy pequeño de datos para el conjunto de entrenamiento (p.ej. 5% entren.; 95% eval.)? ¿y al utilizar un conjunto de evaluación muy pequeño (p.ej. 95% entren.; 5% eval.)?

Una vez concluya la implementación de este procesado, ejecute las secciones necesarias para obtener la matriz de características normalizada correspondiente al conjunto de datos de evaluación (**`features_test_n`**).

A través de la función **`predict_gaussian`**, utilizamos el modelo definido en la fase anterior para predecir la etiqueta de las muestras de evaluación. Abra la función **`predict_gaussian`** y trate de comprender el código que la compone.

> **P17**. Describa brevemente, y de manera conceptual, el funcionamiento de esta función, es decir, el proceso de evaluación de las características de una muestra para predecir su etiqueta.

Una vez se ha efectuado esta predicción, podemos proceder a la evaluación del sistema. Para ello, será necesario comparar las etiquetas predichas (los géneros que nuestro sistema ha discernido) con las originales (los géneros verdaderos).

Una de las maneras más directas de evaluar un sistema es mediante la obtención de la probabilidad de detección y la probabilidad de falsa alarma. Estos dos conceptos se recuerdan brevemente a continuación:

- Pr. Detección: $p_D = \frac{Nº\ muestras\ positivas\ correctamente\ identificadas}{Nº\ muestras\ positivas\ totales}$
- Pr. Falsa Alarma: $p_{FA} = \frac{Nº\ muestras\ negativas\ incorrectamente\ identificadas\ como\ positivas}{Nº\ muestras\ negativas\ totales}$

Compare las etiquetas predichas del modelo completo (**`labels_pred`**) con las etiquetas verdaderas (**`labels_true`**) y calcule las métricas mencionadas anteriormente.

> **P18**. ¿Cuál es el valor de la Pr. Detección? ¿Y el de la Pr. Falsa Alarma? Explique qué representan conceptualmente las dos medidas presentadas para nuestro caso concreto (sistema de discriminación de géneros cinematográficos).

Otra manera más completa de evaluar las prestaciones de un sistema de clasificación como el que se ha desarrollado es mediante el cálculo de una medida llamada Área Bajo la Curva (Area Under the Curve - AUC). El AUC da un valor entre 0.5 y 1 que determina el rendimiento de tu sistema en condiciones generales.

Vuelva a ejecutar la sección actual deteniéndose esta vez en la figura (2) que se genera.

> **P19**. Especifique el valor de AUC obtenido para cada uno de los tres modelos desarrollados. ¿Cuál de los tres modelos es el que mejor rendimiento ofrece en términos de área bajo la curva? ¿Por qué?

> **P20**. ¿Qué implicaría obtener un valor de $AUC = 0.5$? ¿Y $AUC = 1$?