Final Project: Multimedia Classification System

Universidad Carlos III de Madrid, Multimedia, Year 2020-2021

Introduction and goals

The analysis and processing of multimedia data has been a powerful and versatile field of research for many years. One of the many applications of multimedia processing is to extract useful information that can be later used to solve classification problems.

In this project, a movie genre recognition system will be developed. To this end, we will focus on visual features present in the movie posters, as well as on text-based characteristics from their description, synopsys, cast, etc. The database used for this project has been extracted from the well known web platform IMDb[1].

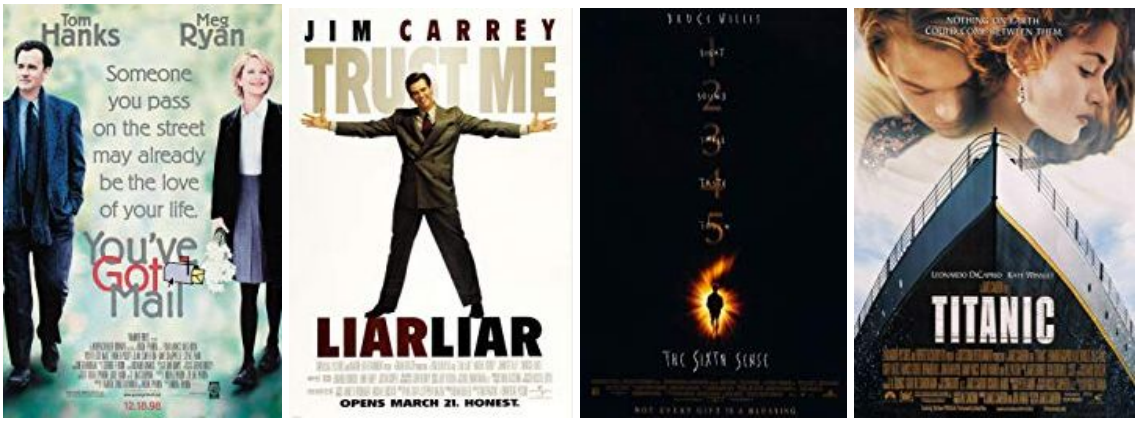

Figure 1. Movie poster examples

The main goals of the project can be summarized in two points:

- Perform an effective visual and text-based feature extraction that properly represents the object under analysis.
- Achieve a decent classification accuracy in the task of distinguishing the movie genres of comedy and drama, based on the extracted features.

Upon the conclusion of this project, the student will have obtained a basic grasp of some of the main basic tools regarding image and text processing.

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

Project delivery ⚠

Important: for this project to be evaluated, it is necessary to upload a compressed file (zip format) to AulaGlobal containing the following:

- A document (**pdf**) following the format set by the file project_template (available in AulaGlobal). The document (compressed in zip) will have a **maximum of 10 pages** and must contain the answers to the questions stated in the Development section. **The use of figures and images to ilustrate and explain the arguments in your answers is strongly recommended**.
- A folder containing all Matlab files used to work. **This folder must work as an independent environment when imported to Matlab**.

  Note: Only one upload is needed for the projects evaluation, i.e., only one of the members of the group has to submit the zip file to AulaGlobal.

Development

The development of the project will be split into 4 stages:

1. Visual feature extraction
2. Text-based feature extraction
3. Training and classification
4. Performance evaluation

For some of these stages, coding will be required, while for others you will just have to run the provided code and reflect on the displayed results.

At this point, you should download the materials related to this project from AulaGlobal. The materials are grouped into 3 main elements:

- A Matlab script (***skeleton.m***) in which you will implement the required code. This script is designed to be completed, and will therefore not work if you run it as it is.
- A library containing useful functions (***lib***) that you will use during development.
- A database (***data***) containing all the necessary images and text data for the system to be properly completed.

Before we begin with the first stage of the project, it is convenient to get used to the data we are going to handle.

Stage 0: Database generation

The first step when starting a new project involving classification systems is the generation of a dataset. In this case, this data is provided as part of the statement, so this step is really simple.

Open the script skeleton.m in Matlab and observe the first section of the code (Read Database). Evaluate just that section.

> **Note**: Remember that you can run a single section by clicking on any line of said section and typing Ctrl+Enter. You can also select parts of the code and turn them into comments with the command Ctrl+R (uncomment using Ctrl+T).

This should leave your workspace with 6 variables, corresponding to the data we will use to discriminate the different cinematographic genres. In particular, we will have two sets of data: the training set and the test set. For each set we will have a variable X, containing the samples (movie posters and descriptive texts) and a variable Y, which contains a label (a number between zero -comedy- and one -drama- that defines the genre) for each sample.

> **Note**: You may interact with these variables and observe their contents by double-clicking their name in your workspace.

Try to click on every variable and access their fields before carrying on to the next stage.

Stage 1: Visual feature extraction

The main goal of feature extraction is to find a way to effectively represent complex data. For this stage, we will focus on the movie posters: images with size 268x182 pxl and 3 color channels (RGB). We can see that, even in cases like this, where image resolution is not particularly great, we need thousands of values to represent a single sample.

We intend to represent each given sample with just 3 values. We will do so through the use of image processing tools. The trouble resides on properly selecting which features to extract and how to extract them so that they are decisive in distinguishing cinematographic genres. Therefore, we will extract features regarding (1) the changes in the color, (2) the brightness and (3) the amount of information present on the edges of the analyzed image.

Changes in color

When working with color features, the HSV color space is a very powerful ally. Particularly, one of its channels (Hue) will be used for this feature. We will try to figure out how high is the variability of this channel. To this end, we will make use of the concept of entropy. Please fill in the blanks to complete the required code in order to identify the Hue channel and calculate the feature as the entropy of the values of said component. You can make use of the functions **`rgb2hsv`** and **`entropy`**.

> **Q1**. What are the advantages that the HSV color space provides over the RGB?

Brightness

Light intensity within a picture gives us a nice approximation of the content of said image. For instance, for outdoors photographs, light can give you an idea about the time the picture was captured. Here, we will be making use of Matlab's function **`mean`** to extract the brightness feature as the mean intensity of the image's Value channel (HS**V**).

> **Q2**. By now, you have made use of the Hue (H) and Value (V) channels. What does the remaining channel represent? Describe -using your own words- the usefulness of said channel.

> **Q3**. Look through your database for two movie posters, one with high brightness and one with low brightness. Show both examples and explain the differences between them (conceptually).

Edges

The gradient of an image can be used, among other things, to extract information regarding the contour of the elements that are present in the image. Use the function **`edge`** to extract -through Sobel method- the edges of the posters. Subsequently, extract the third and last visual feature as the total amount of pixels that belong to an edge.

> **Q4**. For this project, all images have the same exact resolution. If that were not the case, would this feature be useful as it is? Explain why and, if your answer was negative, explain how this problem could be solved.

At this point, you should have 3 values effectively representing each sample (a 268x182 color image representing a movie poster).

> **Q5**. Assuming all values consume the same amount amount of memory, what is our system's savings ratio (percentage of reduced information) with respect to the input data.

As we have seen, our input data consists of images representing movie posters. For the selected visual features, there is no need to apply any preprocessing algorithms over the images. However, some techniques like segmentation or mathematical morphology are often used to prepare our images for a more efficient feature extraction.

> **Q6**. Think of an application of segmentation techniques (particularly, Otsu's method) over the images of our database. Justify your suggestion.

Stage 2: Text-based feature extraction

One of the main advantages of descriptor-based classification systems is that, once the features are extracted, they are but simple values representing a certain sample. For this reason, we are able to take different types of information (as long as they represent the same element) and mix them into a single classification system.

Regarding this particular scenario, we have a descriptive text file to supplement each movie poster. This file has been directly extracted from IMDb -in particular, from the tagless HTML file of each movie page- and it is therefore a raw file that will need some preprocessing before it can be properly handled by any classification system.

This type of processing encompasses a field called Natural Language Processing (NLP) and will be introduced in depth during the second part of this subject. Nonetheless, we will now proceed to extract two very simple features based on the text document.

> **Q7**. Given the current configuration of the system, how many elements will our training feature matrix have once we extract these two new features? Justify your answer.
>
> **Q8**. Assuming that all text files from the training set occupy 5.5 MB and also that we need 8 B to store each element -i.e. each position- of the feature matrix (regardless of the feature), and 24 B to store each image pixel as a whole -i.e. the 3 color components included-. What is our complete descriptive system's savings ratio?
>
> **Note:** Consider 1 MB = 10^6 B

For this first approach, we perform a few simple NLP algorithms to separate the text document into individual words through the use of the function **`obtain_word_array`**. Therefore, you can consider the resulting variable (**words**) as a vector containing the words belonging to the description of each movie.

> **Q9**. Consider the movie Made of Honor (see its poster in Fig. 2). Additionally, consider the following list of words randomly extracted from the corresponding **words** vector:
>
> - the
> - to
> - maid
> - Romance
> - of
> - wedding
>
> Answer the following three questions. From the provided list, which ones will be the most repeated within the words vector? Which ones will be the most repeated throughout the whole corpus of documents? Which ones will be more useful when distinguishing the movie's genre? Justify your answers.

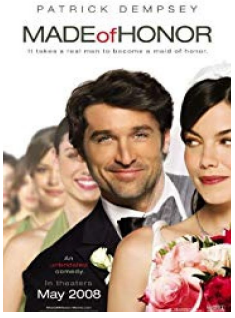

Figure 2. Poster from the movie: Made of Honor (2008).

Complete the required code to select each text document from the training data and extract the text-based features as the amount of words in each document and the mean length of said words. You can make use of Matlab's functions **`length`** and **`strlength`**.

Stage 3: Training and classification

If the feature extraction stage has been correctly implemented, you should now have a variable called **`features`** in your workspace,consisting of 960 rows -one for each training sample- and 5 columns -one for each extracted feature-. However, if we analyze the mentioned variable, we can observe that the range of values for each feature varies wildly. This could be harmful to the system, since a classification algorithm could give greater weigths to certain features, making other insignificant regardless of their actual usefulness. For this reason, it is important to normalize the features before proceeding to train the classification system.

Implement the normalization of the variable by columns, i.e., normalize each feature separately. Remember that the formula used to normalize a vector x is the following:

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

with μ_x and σ_x being the mean and standard deviation of vector x, respectively. You may use Matlab's functions **`mean`** and **`std`**.

You can check if the normalization was correctly performed making use of the function **`check_normalization`**.

Let us stop for a moment at this point and reflect on the usefulness of the extracted features. In the provided skeleton, you can find a section devoted to feature visualization. Through that piece of code, you may choose to display (two at a time) the values of the features for all training samples via scatter plot. Try to display the different combinations of the first 3 features (visual features).

> **Q10**. Based on these scatter plots, which features (or combination of features) do you consider to be more useful regarding the task of genre discrimination? Justify your answer and **use figures** to support it.

Q11. Taking into account the current state of the project's development and the gained knowledge, mention at least 2 additional features (visual or text-based) which could be extracted for this system. Justify your choices.

Once we have the normalized feature matrix, we may proceed to train the system. The main goal is to generate a model using the extracted features and the knowledge we have about each sample (the labels identifying each genre). This model must be able to classify future unknown samples into the two proposed categories (comedy and drama).

We will use a simple Gaussian classifier to train the system. As we are handling a binary classification problem (only two classes are present), we will consider one of them as the positive class (category to identify) and the other as the negative class (category to avoid). This nomenclature is quite useful when dealing classes which are clearly exclusive classes. For instance, when analyzing medical images: cancer (positive class), no cancer (negative class). In this case, however, the choice of the positive class is merely anecdotal and, therefore, arbitrary. Open the function **`fit_gaussian`**, located in the **lib** directory, and try to understand the provided code.

Q12. Let us imagine now that our matrix had just one column, i.e. only one feature was extracted. What will be the size of the covariance matrix extracted by the function **`fit_gaussian`**? Justify your answer.

Q13. Assume the scenario from the previous question. What condition must hold in order for the decision boundary of the classifier to be at equal distances from the mean of both classes at all times? Justify your answer.

Our classification system is now complete. As you can see, we have trained three classification models: (1) using all extracted features, (2) using only visual features and (3) using only text-based features. However, this is useless unless we can provide any numerical confirmation that our system is working properly. To this end, we will see how our system works with samples that has never seen before.

Stage 4: Performance evaluation

Let us remember that, before we started processing the data, we separated it into two different sets: the training set -devoted to obtain our classification model during the previous stages of this project- and the test set. In this stage, we will use this set to evaluate the performance of our system.

Test samples must be subject to the exact same processing that was applied to the training samples. Complete the code for the feature extraction and normalization of the test images.

Note: You may reuse code from previous sections.

Remember that, regarding normalization, you must not recalculate the mean and standard deviation values. The values that were obtained during the previous stage must be reused now.

Q14. Why are we forced to reuse the same normalization parameters from the training stage?

Q15. Why can't we use the same set (e.g. the whole available database) for traning and then again for testing?

Q16. Assuming the size of the database does not change, what is the risk of using an excessively small training set? How about an excessively small test set?

Once this process is implemented, run the corresponding sections to obtain the normalized feature matrix from the test set (**`features_test_n`**).

Through the use of the function **`predict_gaussian`**, we can use the model obtained in the previous stage to predict the label of each test sample. Open said function and try to understand its code.

Q17. Describe (briefly and conceptually) the internal behaviour of this function, i.e., the process followed to evaluate the features of each sample and predict its label.

Now that we have the predicted label for each sample, we may evaluate the performance of the system. To this end, we will compare these predicted labels (the genres our system thinks are correct) with the originals (the true genres of the movies).

One way to get an idea of the performance of a classification system based on predictions is through the Probability of Detection and the Probability of False Alarm. These concepts are briefly reminded subsequently:

- Probability of Detection: $p_D = \frac{Nº\ positive\ samples\ correctly\ identified}{Nº\ positive\ samples\ in\ total}$
- Probability of False Alarm: $p_{FA} = \frac{Nº\ negative\ samples\ incorrectly\ identified\ as\ positive}{Nº\ negative\ samples\ in\ total}$

Compare the predicted labels from the complete model (**`labels_pred`**) with the true labels (**`labels_true`**) and compute the aforementioned metrics.

Q18. What is the value of p_D? What is the value of p_{FA}? Explain what do this metrics represent for the studied scenario (cinematographic genre discrimination).

Another way to evaluate the performance of the system is through the Area Under the Curve (AUC). The AUC is a value in the range 0.5-1 that determines the performance of your system with different possible configurations.

Run the current section again and observe the figure (2) that is displayed.

Q19. Specify the value of the AUC obtained for each of the three models. Which model offers the best performance regarding its AUC? Why?

Q20. ¿What would a value of $AUC = 0.5$ imply? How about $AUC = 1$?