Lab session 6

Audio analysis and feature extraction

1. Introduction

Audio analysis has a wide range of applications in different fields, like entertainment (audio editing, identification), biology (monitor endangered species through audio ID), health care (early detection of diseases like Parkinsons), psychology (audio emotion recognition), and many more.

Audio classification systems often follow a similar structure to image classification ones:

1. Data gathering
2. Data preprocessing
3. Feature extraction
4. Classification

Therefore, a classification model needs to be trained using certain features that represent the samples (in this case: audio signals).

This lab exercise will focus on the third step of the audio classification system, i.e., feature extraction. Incidentally, this step can be reused for other types of audio analysis tools, such as audio matching techniques, since it basically consists in representing the audio signal through different descriptors.

2. Project description

In this lab session, we will attempt a binary classification problem consisting of two classes: cats and dogs. In other words, we will use recordings of cats and dogs to train a system that will later be expected to hear a sound of either a cat or a dog and correctly classify it into the correct animal source.

2.1. Database description

The provided database consists of 203 recordings of cats and dogs (with varying lengths and characteristics). This database has been extracted from Kaggle:

- *https://www.kaggle.com/mmoreaux/audio-cats-and-dogs*

This dataset needs to be divided into a training set and a test set. The optimal percentage of data devoted to each process is left as a choice for the students to build their model.

2.2. Feature extraction

This is the key stage of the process, since it is where the students will perform all the implementations for the basic project requirements. The feature extraction stage is devoted to representing the audio signal using different techniques.

For this stage, let us remember that in order to analyze an audio signal, we often divide this signal into different frames. There are several window types that can be employed to extract the frames (e.g., hamming).

A basic template is provided where 2 basic audio features are extracted:

- Average value of the entropy of the energy of the audio signal's frames.
- Maximum value of the entropy of the energy of the audio signal's frames.

Therefore, in this template, a Nx2 feature matrix is generated (where N is the number of samples that are fed to our feature extraction module). Additional features can be extracted to increase the size of the feature matrix. To this end, the features studied in class can serve as a reference for the students. Some examples of potential features to be extracted are listed below:

- Spectral entropy of the audio signal
- Spectral flux of the audio signal
- Zero-crossing rate
- Mel-frequency Cepstral Coefficients
- etc.

The obtained features need to be then normalized and fed to the classification system to train the model.

2.3. Classification

This stage fits the model with the training data using a Support Vector Machine with RBF kernel. No implementation is required in this stage (unless the specified extra work is performed).

Note that we have specified a positive class (label = 1) for the class 'dog'. However, this decision is completely arbitrary, since this binary classification problem faces two antagonic classes with equal weight. In other words, it differs from a problem where "something to detect" is clear and unambiguous (e.g., 'disease' vs 'no disease').

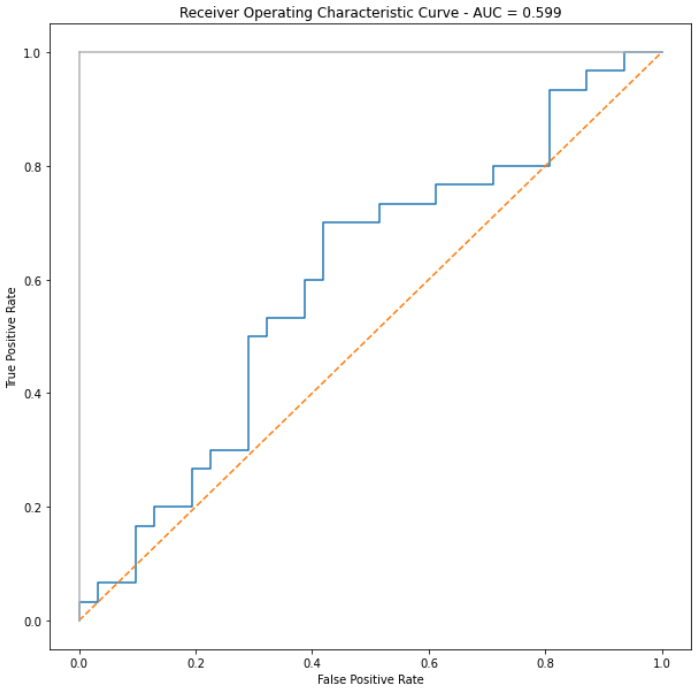
After the model is fitted, we use our test data to evaluate its performance.

2.4. Evaluation

The test data is fed to the system and we obtain a set of predicted scores as an output. These scores represent the probability of each sample belonging to the positive class (the probability of the audio source to be a dog).

In order to evaluate the system without establishing a point of operation (i.e., a threshold for the scores that would yield hard binary labels), we make use of the Receiver Operating Characteristic (ROC) curve. In order to have a numeric representation of this performance, we observe the area contained under this curve as a reference (AUC).

Try running the template with the default features specified in the previous section to visualize this curve. This action should provide a figure similar to the following:



Note that the curve is quite close to the diagonal and, therefore, its AUC is nearing its minimum possible value (0.5). This can be considered as a baseline system to beat.

3. Project requirements

All students are required to provide a basic version of the project. Once this version has been attained, some extra work options may be explored. All students must provide a brief report to describe their implementation (see Section 3.3)

3.1. Basic project

A minimum of three additional features should be implemented for the feature extraction stage. To this end the students will make use of the provided custom library audio_features.py (available within the materials in AulaGlobal)

Note that the use of **scatter plots** to compare and analyze the extracted features is highly advised. Finally, take into account that your model will be evaluated using previously unseen samples, so try to avoid overfitting.

3.2. Extra work

This part is **optional**, and can add up to +0.5 points to the continuous evaluation of the subject. Lines of work to explore include:

- The implementation of two additional features, optionally making use of the librosa library for audio analysis, whose documentation can be found in the following website:
  - https://librosa.org/doc/latest/index.html
- The implementation of a mechanism to select or extract the most relevant features of the system.
- The design and implementation of an automated validation process to find the best values for the parameters of the extracted features' functions.

3.3. Delivery

The submission of this lab must consist of the following:

1. The **software** implemented according to the Python scripts provided, properly structured and commented. If you have used some source of external software, you have to include it in your submission (and properly describe it in the project report).
2. A brief project **report** (2 pages maximum, excluding the cover; 3 pages if you do extra work) including the following information:
   a. Brief introduction (one paragraph; it is just a formality)
   b. Feature extraction: describe the extracted features and justify your choices.
   c. Evaluation and discussion of results
   d. References

The report must be brief but precise in terms of the description of the procedure used. The student must provide all relevant information about the system and the experimental protocol (with the exception of the description of the known methods or algorithms – those that can be found in papers and books), so that a reader of the report could reproduce the experiments.

4. Project evaluation

The evaluation of this lab is structured as follows:

- 80%: technical content of the report.
- 20%: results and quality of the code.

All original contributions will be valued positively. The instructor will evaluate the design of the segmentation system. Failed approaches will also be considered as long as they are properly motivated and described.