Statistical Machine Learning

Hilary Term 2024

Group-Assessed Practical

Music Genre Classification

Description. This project aims at predicting the musical genre of a song, based on some pre-computed features. The dataset consists of $p = 518$ pre-computed features extracted from 8000 audio tracks. Each song i is represented by an input vector $x_i = (x_{i1}, \ldots, x_{ip})$ where $x_{ij} \in \mathbb{R}$ represents the j'th feature of song i. The features are real-valued and correspond to summary statistics (mean, standard deviation, skewness, kurtosis, median, minimum and maximum) of time series of various music features, such as the chromagram or the Mel-frequency cepstrum. Each song may be of eight different classes: Electronic, Experimental, Folk, Hip-Hop, Instrumental, International, Pop or Rock. The objective is to (i) construct a classifier which, based on the features of a song, predicts its genre, and to (ii) estimate its generalisation error under the 0–1 loss.

The dataset is split into a training set of 6000 audio tracks, and a test set of 2000 audio tracks. For the training observations, you have access to both the inputs (X_train) and outputs (y_train). For the test set, you have only access to the inputs (X_test), and the objective is to predict their genre.

Methods. You are free to use any machine learning technique you wish, as long as you describe clearly in the report all the steps and choices that you have made. While getting a good predictive performance for your method will be important, remember that you will be assessed based on the quality of your report; so explaining your steps and choices clearly and discussing all the issues you have faced in this practical will be essential. Besides explaining your final classifier, you should also describe some of the other techniques you have tried and include a brief description of the more computational aspects of your work. It is particularly important to discuss the potential advantages/disadvantages of the different methods considered, in terms of interpretability, computational cost, etc. You can use any programming language you wish (Python, R, etc.), and any available library/toolbox, as long as you understand and can describe the methods used. In Python, most of the methods covered in the course (except convolutional neural networks) are implemented in Scikit-learn. In R, many machine learning methods are implemented in the (meta)-package caret.

Report. The report has a limit of 2,500 words. Please be as concise as you can. You should work in teams of 4 participants. Remember to place your team name, which consists of the collated anonymous IDs of all group members, on the cover page of the report. Please include the code you used to get your final score as an appendix (this is not counting towards the 2,500 words limit). Make sure the code is readable (i.e. it contains comments explaining what you are doing). Only one student from each group is required to make the submission.

François Caron: caron@stats.ox.ac.uk

Submissions. Together with your report, you should also submit a csv file, containing the predicted class for the 2,000 observations in the test set. Your report should include an estimate of the classification accuracy on the test set of your prediction. The submission file (csv format) should contain two columns: Id and Genre. The file should contain a header, followed by the 2,000 class predictions, and have the following format:

```
Id,Genre
0,Hip-Hop
1,Pop
2,Rock
...
```

A sample submission file is available.

Files available:

- This pdf with the instructions
- Training inputs: `X_train.csv`
- Training outputs: `y_train.csv`
- Test Inputs: `X_test.csv`
- Sample submission file: `myprediction.csv`
- Sample Python code: `AssessedPracticalSamplePythonCode.ipynb`

Evaluation metric. The metric used is the classification accuracy (proportion of well-classified examples in the test set).

Sample Python code. A sample Python notebook is provided. The code loads the data, fits a k-nearest neighbour classifier on the training set and predicts the class in the test set. It then exports a csv file of the correct format. For your information, the 5-nearest neighbour classifier has a classification accuracy of about 35% on the test set. You can use this value as a benchmark, and a lower bound for the performance of your classifier; you should be able to achieve higher performances with other methods.

Deadline. The deadline to submit your pdf report and csv file is Wednesday 20 March noon (week 10).

François Caron: caron@stats.ox.ac.uk

Additional information about the features.
This section is just provided for your information. There is no expectation that you should have a good understanding of how the pre-computed audio features were obtained.

For each feature, the values have been extracted over different time frames using the library **librosa**. Each time series has then been summarised with seven summary statistics (kurtosis, min, max, mean, median, skewness, standard deviation). For more information about the musical features, see:

`https://librosa.org/doc/latest/feature.html`

Name	Description	Number of features
chroma_cens	Chroma Energy Normalized (CENS, 12 chroma)	7*12=84
chroma_cqt	Constant-Q chromagram (12 chroma)	7*12=84
chroma_stft	Chromagram (12 chroma)	7*12=84
mfcc	Mel-frequency cepstrum (20 coefficients)	7*20=140
rmse	Root-mean-square	7
spectral_bandwidth	Spectral bandwidth	7
spectral_centroid	Spectral centroid	7
spectral_contrast	Spectral contrast (7 frequency bands)	7*7=49
spectral_rolloff	Roll-off frequency	7
tonnetz	Tonal centroid features (6 features)	7*6=42
zcr	Zero-crossing rate	7

François Caron: caron@stats.ox.ac.uk