Statistical Machine Learning

Hilary Term 2021

Group-Assessed practical

Description. This practical has an associated Kaggle-in-class challenge. The dataset consists of $p = 518$ pre-computed features extracted from 8000 audio tracks Tracks were obtained from the Free Music Archive: https://freemusicarchive.org/

The pre-computed features are real-valued and correspond to various music features (such as the chroma feature, or the Mel-frequency cepstrum) computed over different windows, and summarised by some summary statistics (mean, standard deviation, skew, kurtosis, median, minimum and maximum) within each window. There is no need here to understand what these features represent.

The dataset is split into a training set of 6000 audio tracks, and a test set of 2000 audio tracks. To each song in the training set is associated one of eight musical genres: Electronic, Experimental, Folk, Hip-Hop, Instrumental, International, Pop, Rock.

The objective is, based on the pre-computed features of the audio tracks in the test set, to predict their genre.

To access the Kaggle challenge, and download the data, follow this link:
https://www.kaggle.com/t/9292af3b50984641a3b234bba0208c9d.

Methods. You are free to use any machine learning technique discussed in the course, as long as you describe clearly in the report all the steps and choices that you have made. While getting a good predictive performance of 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 challenge will be essential. Besides explaining your final solution, please briefly describe some of the other techniques you have tried and include a brief description of the more computational aspects of your work. 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 name the pdf file of your submitted report using the list of anonymous IDs of all team members, e.g. P001-P002-P003-P004.pdf. 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

François Caron: caron@stats.ox.ac.uk

comments explaining what you are doing). Please indicate in the report the name of your Kaggle team (or name your team using the anonymous IDs of the team members).

Submissions. The evaluation of your methods will be made via the Kaggle-in-class website. You should create a single anonymous Kaggle account per team (do not create multiple accounts). You can make up to 5 submissions every day.

Submission files (csv format) should contain two columns: Id and Genre. The file should contain a header, followed by the 2000 genre predictions, and have the following format:

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

A sample submission file is available on the website.

Evaluation metric. The metric used is the classification accuracy (proportion of well-classified examples in the test set). When submitting your predictions on Kaggle, you will instantly see the accuracy of your method on approximately 50% of the test data (called the public leaderboard). At the end of the competition, the final results will be calculated on the other 50% (private leaderboard). Teams can hand-select the eligible submission for the final (private) ranking, or will otherwise default to the best public scoring submission.

Sample Python code. A sample Python notebook is available. The code loads the data, fits a Naive Bayes model to the training data and predicts the genres in the test set. It then exports a csv file of the correct format, to be uploaded on Kaggle. The Naive Bayes predictions also give you a benchmark to which you can compare your results. (Note: the example notebook is hosted by Kaggle, but you do not need to run your experiments on Kaggle. You can run them on your machine, and upload your predictions when ready.)

Help on Kaggle-in-class. You will find here a FAQ on Kaggle in-class challenges.

Deadline. The deadline to submit your report is Wednesday 24 March 12:00pm (week 10).