MLBP 2018 Data Analysis Project

The problem

The data analysis project involves the design of a complete machine learning solution. In particular, the project revolves around the task of identifying the music genre of songs. This is useful as a way to group music into categories that can be later used for recommendation or discovery. The problem of music genre classification is difficult: while some genres distinctions are fairly straightforward (e.g. heavy metal vs classical), others are fuzzier (e.g. rock vs blues).

In this data analysis project, your task is to construct a predictor h(x) for each genre Y, which takes the features x and maps it to a probability h(x) that the genre is "rock" (e.g.) or not. You should try out different machine learning methods, including but not limited to those presented throughout this course, for predicting the music genre of songs. The dataset which is provided to solve this task contains preprocessed audio information. In particular, the raw audio signals have been transformed to carefully chosen features.

The data

The data is split into two datasets: a training data set with 4363 songs, and a test set dataset with 6544 songs. Each song has 264 features, and there are 10 possible classes in total. The dataset is a custom subset of the Million Song Dataset, and the labels were obtained from AllMusic.com. For simplicity, each song has been assigned only one label that corresponds to the most representative genre. The 10 labels are:

1 'Pop_Rock'
2 'Electronic'
3 'Rap'
4 'Jazz'
5 'Latin'
6 'RnB'
7 'International'
8 'Country'
9 'Reggae'
10 'Blues'

The features provided are a summary representation of the 3 main components of music: timbre, pitch (melody and harmony) and rhythm. A very brief description of these components:

Timbre: "The tonal colour, or timbre, is a multidimensional psychoacoustic measure. When two sounds have the same pitch, loudness, and duration, timbre is what makes one particular musical sound different from another. For example, the same musical notes played by a piano and a trumpet are easily distinguished by listeners as being different. The best

physical explanation for this difference comes from the spectrum and its variation with time." [from Communication Acoustics]

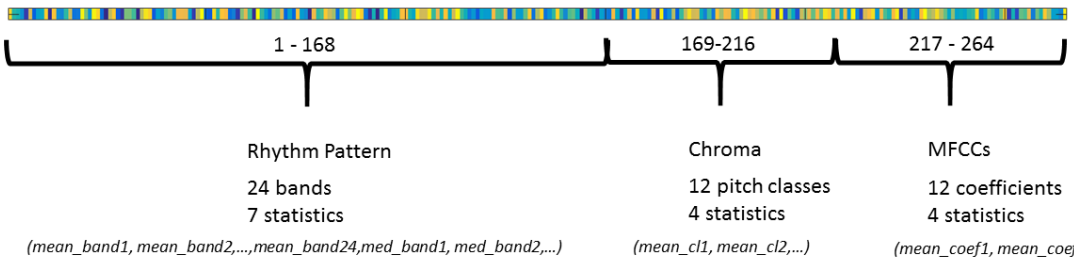
Rhythm: "Rhythm is a complex concept which refers to different temporal structures in music. The existence of rhythm is based on natural repetitions in time, such as walking, running, the heartbeat, and breathing." [from Communication Acoustics]

Pitch: "Pitch is defined by the American National Standards Institute as ‘that auditory attribute of sound according to which sounds can be ordered on a scale from low to high’ (ANSI-S1.1, 2013)." [from Communication Acoustics]

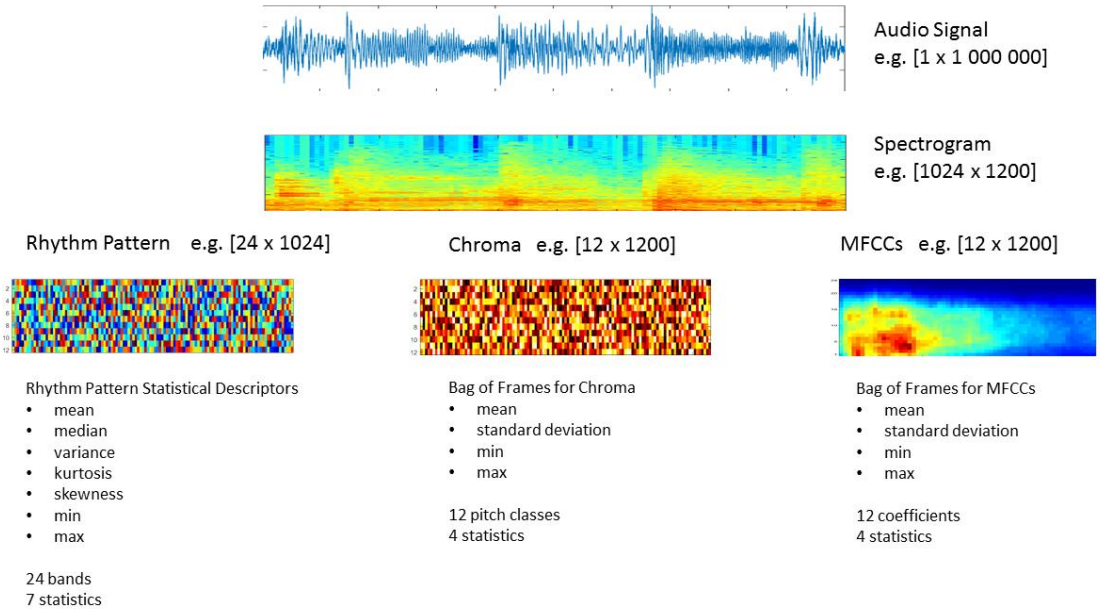
The features used include the first 12 *MFCCs* (Mel Frequency Cepstral Coefficients), *chroma* values, and *rhythm* patterns. In short, the MFCCs provide a spectrum of the spectrum of an audio file, showing the overall shape of the frequency content and can be used to describe timbre. The chroma features are a condensed representation of the audio spectrum, where the entire spectrum is projected onto 12 bins representing the 12 distinct semitones (or chroma) of the musical octave used in most western music. Thus the chroma can be used to analyze pitch (melody and harmony). Finally, the rhythm patterns describe how much modulation is happening on certain ranges of the frequency spectrum.

During the preprocessing, because music is a time dependent signal, all features were extracted using a frame by frame decomposition of the signal. For the project, the features are then condensed using the bag of frames model, where some summary statistics were computed for each feature set. The final feature vector of each song consists of 264 dimensions: 168 values for the rhythm patterns (24 bands, 7 statistics), 48 values for the chroma (12 bands, 4 statistics), and 48 values for the MFCCs (12 bands, 4 statistics).

Features vector of each song:



Feature extraction process:



File descriptions

- **train_data.csv** - the training data set, providing 4363 samples described by 264 features.
- **train_labels.csv** - the labels of the train samples.
- **test_data.csv** - the 6544 test samples of which the labels should be predicted. Don't change the order of the samples, as your predicted labels should be ordered accordingly.
- **dummy_example_solution.csv** - a sample submission file in the correct format, used as a benchmark solution.

Your submission should be in the following format (also shown in the dummy_example_solution.csv file):

For the competition using accuracy, the submission format should be:

First column:

- **Header**: Sample_id
- **Data**: Succeeding integer identifiers (1,2,...,6554)

Second column:

- **Header**: Sample_label
- **Data**: Predicted labels (integers from 1 to 10)

Sample_id	Sample_label
1	1
2	10
3	5
...	...
6544	1

For the competition using multiclass LogLoss, the submission format should be:

First column:

- **Header**: Sample_id
- **Data**: Succeeding integer identifiers (1,2,...,6554)

Second to tenth column:

- **Header**: {Class_1, Class_2, ... , Class_10}
- **Data**: Likelihood that indicates the probability a sample belong to each of the classes, i.e. a matrix of 6544 x 10.

Sample_id	Class_1	Class_2	…	Class_10
1	0.0669	0.0274	…	0.0032
2	0.0449	0.0219	…	0.002
3	0.7304	0.0205	…	0.0326
…	…	…	…	…
6544	0.0154	0.4542	...	0.0033

Hints

- Try different classifiers and methods.
- Take some time to analyze the data (class distribution, features ranges, etc.).
- There might be some redundancy in the data, so explore dimensionality reduction or feature selection.
- Try semi supervised learning using the test dataset.
- If something does not work, in the report, elaborate on why it is not working.

Multiclass classification:

Adapting a binary classification method for this multiclass task is very straightforward using the “one vs all” technique. The main idea is to train M binary classifiers where M is the number of classes, and for each classifier you assign 1 to the data samples which label is equal to M, and 0 to everything else.

Then, to predict the class for new data, you need to compare the output of the M binary classifiers. This varies depending on the type of classifier used, but usually you can get the probability of a data point belonging to class M for each binary classifier. The new data should be classified to the class with the highest binary probability.

Performance evaluation

The two evaluation metrics used in the Kaggle competitions are:

- Categorical Accuracy
  - $accuracy = \frac{|y_{true} = y_{predicted}|}{N}$

Where N is the total number of data samples, y_{true} is a vector containing the actual class of all samples, and $y_{predicted}$ is a vector containing the predicted classes for all samples.

- Multiclass logarithmic loss
  - $log - loss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j} log(p_{i,j})$

Where N is the total number of data samples, M is the total number of classes, $y_{i,j}$ is an indicator function that outputs 1 when the sample i is assigned to class j, 0 otherwise; and $p_{i,j}$ is the predicted probability that the sample i belongs to class j. NOTE: in Kaggle, the probabilities of each data sample do not have to sum to 1 because they are normalized in the evaluation process.

https://www.kaggle.com/wiki/LogLoss

You are required to beat a baseline solution on both performance metrics. You are allowed to adopt two different approaches, or go with one and evaluate with both metrics; up to you. You should however include some reflection on the different nature of the two metrics. Why does a solution score high on one metric, but low on the other, what does it tell you. Or (if you adopt two different approaches), why did you make that decision, and relate to the metrics.

The Report

The project report should be written as Python notebook. Follow the general structure of a scientific article but also include the used Python source code. The report should not be too long (less than 10 pages excluding the source code).

Formal structure:

Title - Concise and informative, describes the approach to solve the problem. Some good titles from previous years:

- Comparing extreme learning machines and naive bayes' classifier in spam detection
- Using linear discriminant analysis in spam detection

Some not-so-good titles:

- Bayesian spam filtering with extras
- Two-component classifier for spam detection
- CS-E3210 Term Project, final report

Abstract - Precise summary of the whole report, previews the contents and results. Must be a single paragraph between 100 and 200 words.

Introduction - Background, problem statement, motivation, many references, description of contents. Introduces the reader to the topic and the broad context within which your research/project fits

- What do you hope to learn from the project?
- What question is being addressed?
- Why is this task important? (motivation)

Keep it short (half to 1 page).

Data analysis - Briefly describe data (class distribution, dimensionality) and how will it affect classification. Visualize the data. Don't focus too much on the meaning of the features, unless you want to.

- Include histograms showing class distribution.

Methods and experiments - Explain your whole approach (you can include a block diagram showing the steps in your process).

- What methods/algorithms, why were the methods chosen.

- What evaluation methodology (cross CV, etc.).

Results - Summarize the results of the experiments without discussing their implications.
- Include both performance measures (accuracy and LogLoss).
- How does it perform on kaggle compared to the train data.
- Include a confusion matrix.

Discussion/Conclusions - Interpret and explain your results
- Discuss the relevance of the performance measures (accuracy and LogLoss) for imbalanced multiclass datasets.
- How the results relate to the literature.
- Suggestions for future research/improvement.
- Did the study answer your questions?

References - List of all the references cited in the document

Appendices - Any additional material needed to complete the report can be included here. For example, if you want to keep “Methods” and “Results” sections short, you can explain the winning algorithm there and move the other algorithms that you have tried in the appendix. The content should be relevant to the report and should help to explain or visualize something mentioned earlier. ***You can remove the whole Appendix section if there is no need for it.***

Points and grades:

Project report:	
Title	(6 p)
Abstract	(6 p)
Introduction	(12 p)
Data Analysis	(30 p)
Methods and Experiments	(48 p)
Results	(30 p)
Discussion/Conclusions	(18 p)
References	(6 p)
Appendices	(6 p)
Overall quality	(18 p)
Report total:	180 p

Kaggle:	
Beating the benchmark in both competitions	60 p
(Extra) Data Analysis presentation (top-5 teams)	+30 p

Peer grading	60 p

(see https://docs.moodle.org/35/en/Using_Workshop#Workshop_grading for details)

Total = 300 / 330 p

References

Pulkki, Ville and Karjalainen, Matti, "*Communication acoustics: an introduction to speech, audio, and psychoacoustics",* John Wiley & Sons, 2015.

T. Ganchev, N. Fakotakis, and G. Kokkinakis, "Comparative evaluation of various MFCC implementations on the speaker verification task," in International Conference on Speech and Computer (SPECOM'05), 2005, vol. 1, pp. 191–194.

Mel-frequency cepstrum - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Mel_frequency_cepstral_coefficient

Chroma Feature Analysis and Synthesis https://labrosa.ee.columbia.edu/matlab/chroma-ansyn/

Vienna University of Technology -Institute of Software Technology and Interactive Systems Information & Software Engineering Group, "Audio Feature Extraction - Rhythm Patterns" http://ifs.tuwien.ac.at/mir/audiofeatureextraction.html

Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. "*The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)"*, 2011. https://labrosa.ee.columbia.edu/millionsong/

S. Katzoff, "Clarity in Technical Reporting," NASA report SP-7010, Second Edition, 1964. http://www.ifs.tuwien.ac.at/~silvia/research-tips/NASA-64-sp7010.pdf