Artist Recognition in Music

Alessandro Pedrani: *Università degli Studi di Torino*
Matteo Rossi: *Università degli Studi di Torino*

December 9, 2020

Music information retrieval is an interdisciplinary science, focusing on retrieving information from music, possibly in an automated way. This project concentrates on artist recognition, i.e. we propose two strategies to assign an audio file representing a song to its artist. This is of course of interest for automatic tagging but also for recommender systems; moreover same principles are employed in speaker verification systems used for example in Alexa, Google Assistant and Siri. A lot of works have been published in this direction and the state of art is [2]. However, though statistically beautiful, this is really hard to implement, and almost impossible to run on our computers in a reasonable time. Therefore we developed simpler but still interesting algorithms.

1 Database and Features

Music information retrieval is naturally based on audio, so we judged more appropriate, and more fun, to create our own database of songs from artist that we like. Therefore we selected 55 songs from a variety of albums of each of the 10 following artists: Bruce Springsteen, Coldplay, Ed Sheeran, Guns n' Roses, Michael Jackson, Passenger, Pink Floyd, Queen, Simon & Garfunkel and The Beatles. This leads to a balanced dataset of $N = 550$ songs. Notice how this is a multiclass classification problem with lot of classes and few observations per class (at least with respect to what we saw during the course).

Since every audio file contains a huge amount of informations, we were able to experiment with different features, ranging from high level characteristics like tempo, key, loudness ecc., to low level quantities like MFCCs and Chroma. MFCCs is the acronym for Mel-Frequency Cepstral Coefficients, a frame-by-frame concise representation of the overall shape of the spectral envelope of a sound in the non linear Mel-scale. Their definition and construction is quite complicated, a detailed explanation can be found in [3]. Chroma features instead describe the frame-by-frame energy in the twelve different pitch classes (C,C#,. . . ,A#,B).

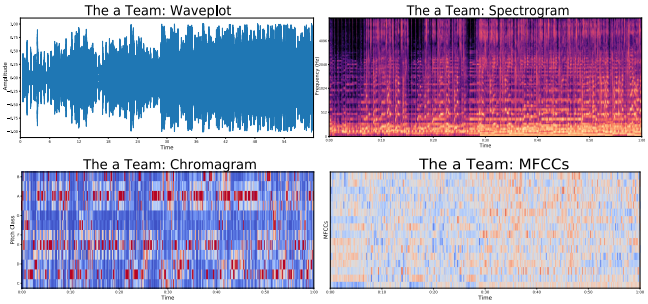

Figure 1: *Visual Representations of a Song*

It turns out that the best choice are MFCCs (though Chroma features can slightly improve performances [1], but at an unfeasible computational cost for our laptops). Therefore we decided to use only MFCCs as features and to study how performances are effected if we change the "quality" of features, that is if we decrease the sampling rate from the optimal value of $10kHz$ and if we decrease the number of coefficients from 20 up to 4.
We underline that since every song has a different duration, then features $\{X_j\}_{j=1}^{N}$ of the songs will be matrices belonging to spaces of different dimensions; namely, the number of rows is the number of coefficients we decide extract (and its the same for all song) and the number of columns depends on the

sampling rate (determining how many *ms* a frame is long) and on the duration of the song (that is out of our control). This require special attention in the organization of features, and makes impossible the direct use of algorithms we saw in the course.

We also point out that MFCCs are low level features so regardless of the interpretability of the models that we will use, using MFCCs means losing every interpretability, but this is necessary because high level features are not able to discriminate between-artist from within-artist variability.
Finally notice that we will work with MFCCs normalized song-by-song instead of globally normalize the MFCCs of all frames of all songs, as they are considered more robust to noise, and more generally to "recording" conditions. Moreover MFCCs show only a little correlation each other (because of the way in which they are constructed), and song-by-song they seem normally distributed.

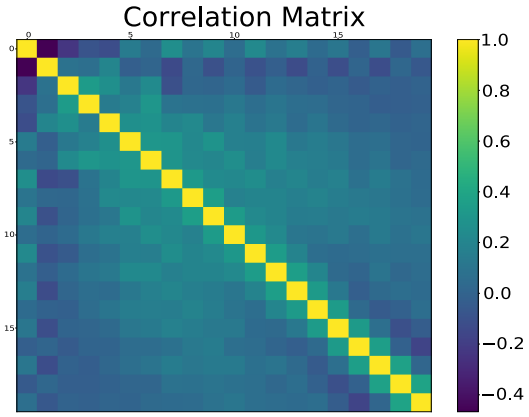

Figure 2: *Correlation Matrix of MFCCs of all frames*

2 Models

Let's discuss our two main methods for accomplishing artist classification. Both are based on two steps:

1) classify frames $\{f_{i,j}\}_{i=1,\dots,n_j}^{j=1,\dots,N}$ to artists using a whatever classifier;
2) classify a song S_j to the artist to which the majority of frames its frames $\{f_{i,j}\}_{i=1,\dots,n_j}$ were classified.

This is a simple idea, that implicitly introduces the assumption that frames are independent, often referred as *Bag-of-Frames* assumption. Notice how this is clearly a bad choice for songs, though is always assumed in literature. Since we haven't found a clever way to use the temporal structure of songs we will unfortunately ride the wave and assume *Bag-of-Frames*.

The first method we propose for step $1)$ is the use of the K-Nearest Neighbors classifier, which is naturally suitable for multi-class problems and requires only the tuning of K; this is an advantage since our problem is so computationally heavy that makes tuning unfeasible. Moreover among "standard" classifier we tried this gave us the best result.

The second method instead uses a (not really) clever "ad-hoc" frame classifier. We model frames of songs of each artist as different Gaussian mixture model: $M_1, \dots M_{10}$ (with a big number of components). Hence for each $a = 1, \dots, 10$ we learn via expected maximization the parameters $(\mathbf{w}^a, \mu^a, \mathbf{\Sigma}^a)$ of the gaussian mixture M_a using frames coming from songs of the a-th artist. Given a new frame f, we can compute the log-likelihood of f given each of the models. Clearly we assign f to the artist whose model gives the highest score. This is therefore a Bayes Classifier for frames were we assume that each artist has the same probability *a priori*; with a forgivable abuse of notation:

$$p(a \mid f) \propto p(f \mid a)p(a) \propto p(f \mid a) = p(f \mid M_a)$$

Finally we proceed to step 2) as before, assigning each song to the artist to which the majority of its frames were classified.

We analysed also possible extensions of this second method. The first is using a Bayesian Gaussian Mixture that learns from the data the number of components to use in the mixture model of each artist. However results are not too sensible to the number of components, as long as the number is not too small or too large, so this provides only a little conceptual improvement. Moreover one could go fully Bayesian and use variational inference for a Gaussian mixture model instead of EM. The other extension is a *frame-selection*. Since our goal is not to classify all frames, then to classify songs we can decide to use only frames for which one of the model predominates on the others in a significant way and label the other as *null*-frames. Interestingly the tuning of the threshold is feasible for us because it doesn't require to re-fit the frame classifier and for the optimal value it really provides an improvement.

3 Results

To evaluate our models we used *micro* and *macro* averaged *F1 scores*, i.e. the harmonic means of *micro* and *macro* averaged *precision* and *recall*; it's easy to prove that micro measures are equal to the accuracy in a multi-class setting. We judged these as suitable for us because we have a balanced dataset and in this framework we are interested in predicting well, without particular care on one of the classes; moreover in music information retrieval these are the most used measures. A comprehensive guide on multi-class performance measure that has been useful for us is [4]. Figure 3 summarize how performances of our two methods vary if we change the quality of features. These are results obtained on the test set, with a single, 75% train and 25% test, balanced split. We decided not to use cross validation error just because of our limited computational power. In practice, we report here for brevity only the accuracy, since micro and macro scores are always really close.

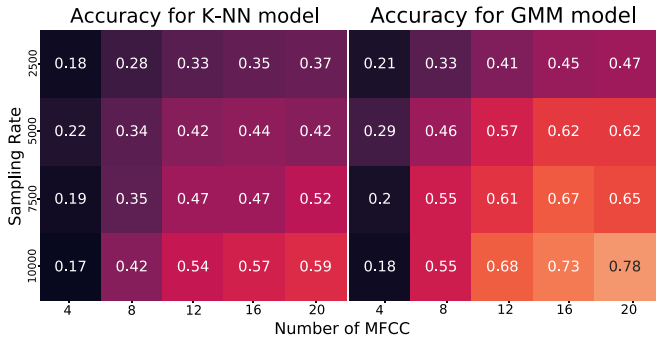

Figure 3: *Summary of the results*

What we can conclude from Figure 3 is that high quality features are really needed to have good performances, and in particular it seems that 4 MFCCs are too poor.

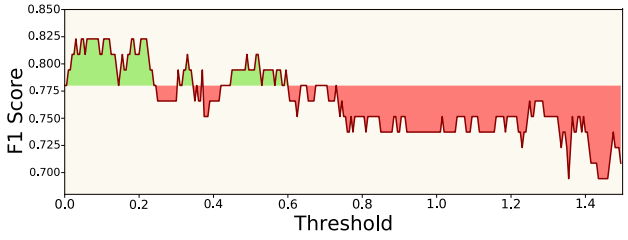

Figure 4: *Tuning of the significance threshold parameter*

Figure 4 instead represents how performances of the GMM model with 20 MFCCs and a sampling rate of $10kHz$ vary, if we change the required significance threshold[1]. It shows that if we require it to be $c = 0,03$, we will improve performances and get an accuracy of $0,825$. However we don't consider it a reliable improvement, since results tend to oscillate a lot as c varies. Hence we report the test confusion matrix for the more stable best standard GMM.

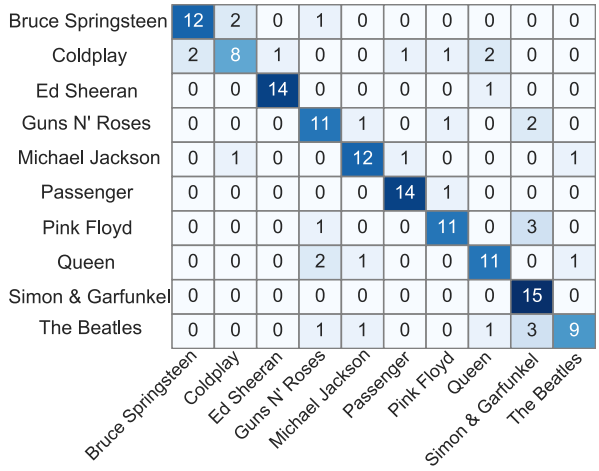

Figure 5: *Confusion matrix for the best model*

We conclude by being honest: these results maybe be a little bit inflated by the so called *producer effect*. Indeed to train our models we have randomly selected songs of each artist instead of doing an album-based split, and since songs by an artist tend to be more similar within the same album then across different records, splitting according to entire albums can decrease performances.

Bibliography

[1] Daniel Ellis. *Classifying Music Audio With Timbral and Chroma Features*. 2007.

[2] Markus Schedl Hamid Eghbal-zadeh and Gerhard Widmer. *Timbral Modelling for Music Artists Recognition Using I-Vectors*. 2015.

[3] S.A. Samad T. F. Idebeaa S. Majeed H. Husain. *Mel Frequency Cepstral Coefficients. Feature Extraction Enhancement in the Application of The Speec Recognition: A comparison Study*. 2015.

[4] Marina Sokolova and Guy Lapalme. *A Systematic Analysis of Performance Measures for Classification Tasks*. 2009.

[1] The significance threshold is the difference between the best and the second-best GMM that we require in order not to classify the frame as null.