AUDIO CLASSICAL COMPOSER IDENTIFICATION IN MIREX 2015: SUBMISSION BASED ON STRUCTURAL ANALYSIS OF MUSIC

Leopoldo Pla Sempere
Polytechnic University of Valencia
leoplsem@posgrado.upv.es

ABSTRACT

Based on musicological fundamentals of classical composers and the structural analysis primary ideas, in this work I study some compositive features to train a neural network using author recognition techniques applied to the task of Audio Classical Composer Identification in MIREX 2015.

1. INTRODUCTION

I propose a model based on some of the high level features that the humans use to identify a composer in an audio or a score: musical structure and harmonic analysis. These features are parallel to the features that is usual to look for in a text, as POS tagging, n-grams, different kind of element counts (number of punctuation marks, hyphens, number of paragraphs, etc.) or the general structure of the text (introduction, development of the text, conclusion, etc.).

These features are susceptible to having errors. However, as shown in the PAN contest for author identification [**?**] and author profiling [**?**] the previous features discriminate the different authors with a similar accuracy that the state of the art of the Audio Classical Composer Identification task [**?**].

2. FEATURE EXTRACTION

Some of the next explained features are implemented by myself and other ones are implemented through the Queen Mary plugins of the Centre for Digital Music (University of London), tested with Sonic Visualizer [**?**] and applied to the dataset using Sonic Annotator [**?**].

2.1 Key Mode

The Vamp Plugin ”Key Mode” calculates the major or minor mode of the estimated key in windows of 10 chroma frames. After calculate them, I use the count of the changes between minor and major as a feature.

2.2 Segmentation

This is also feature from a Vamp Plugin from Queen Mary which divides the channel into 10 structural segments based on Chroma and MFCC. Also, it labels similar segments, which gives us a structural analysis of the sample based on tonality. As key mode, I also use the count of the segments that appear at the segment.

2.3 Tonality

The main work has been done in these following high level features based on the key of the sample and the detected chords. First of all, I obtain through the key strength Vamp plugin the value between -1 and 1 of every key (from C, C#, and so on, to B minor and major) of every window of 1 chroma frame. Then, I select the most strong key of every window.

This set of windows, then, need to be transposed to convert this incontextual chords into a functional chords to get a functional harmonic analysis of the sample. To transpose, I use the key of the sample, which I obtain using a weighted sum of the number of perfect cadences found at the key strength using 1 chroma, and the values of key strength plugin using 10 chroma. This key is used as a feature.

2.4 Unigram, bigram and trigrams from harmonic analysis

From the previous feature we can obtain the number of functional units in the sample (tonic, dominant, subdominant, etc.), the number of most used cadences (perfect authentic cadence, plagal cadence, half-cadence, etc.) and even a set of most used progressions of three chords (like IV-V-I). As a parallelism of the POS tagging in text, is interesting to analyse the impact of this features because it's known that discriminate the composers [**?**].

2.5 MFCC means

At last, I included the means of the MFCC (Vamp Plugin) from the sample, using 20 coefficients and including C0, also to compare this low level feature with the previous ones.

3. CLASSIFICATION

In the proposed systems, I used a Neural Network (NN) in one system and a Deep Belief Network (DBN) in another

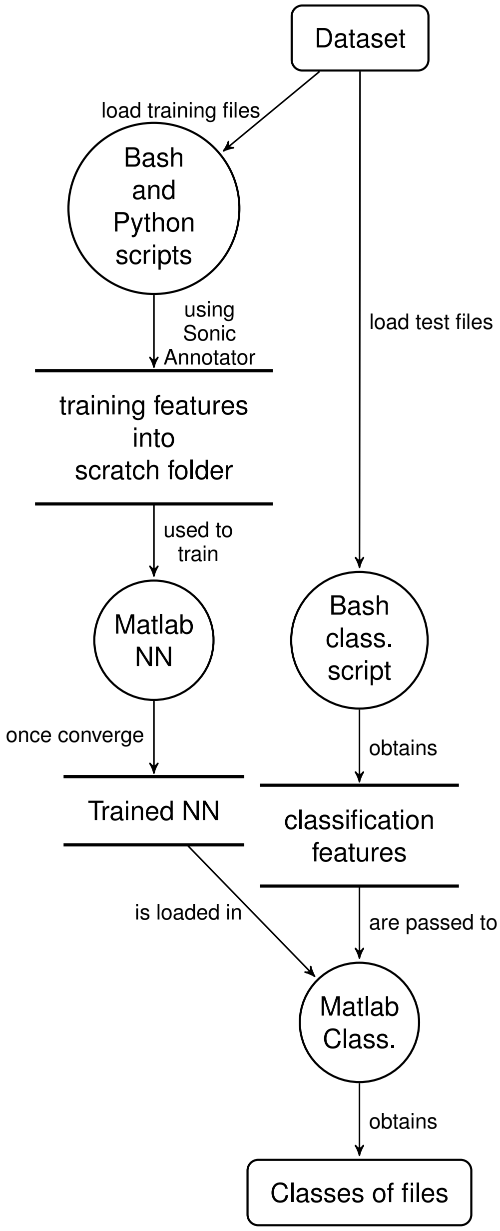

Figure 1. Workflow of the proposed system

one to pre-train a Neural Network as a classifiers implemented by Rasmus Berg [**?**]. The features are normalized with z-score and after that, normalized between 0 and 1 before using the neural network. The networks are configured with 44 neurons at the hidden layer, 300 epochs, a batch size based on a divisor of the number of dataset samples, sigmoid activation function and softmax function at the output layer. I tested the system with a homemade database of FLAC files, extracted from my own CD's of the authors of the task.

4. ACKNOWLEDGEMENTS

This research has been developed as a Master's degree final project at the Department of Computer Systems and Computation of the Polytechnic University of Valencia.

5. REFERENCES

[1] Efstathios Stamatatos, Walter Daelemans, Ben Verhoeven, Martin Potthast, Benno Stein, Patrick Juola, Miguel A. Sanchez-Perez and Alberto Barrn-Cedeo: "Overview of the Author Identification Task at PAN 2014", *Bauhaus-Universitt Weimar*, pp. 10–16, 2014.

[2] Francisco Rangel, Paolo Rosso, Moshe Koppel, Efstathios Stamatatos, Giacomo Inches: "Overview of the Author Profiling Task at PAN 2013", *Bauhaus-Universitt Weimar*, pp. 6–10, 2014.

[3] J. Stephen Downie and IMIRSEL: "MIREX 2014 Evaluation Results", *University of Illinois at Urbana-Champaign*

[4] Chris Cannam, Christian Landone, and Mark Sandler: "Sonic Visualiser: An Open Source Application for Viewing, Analysing, and Annotating Music Audio Files", *Proceedings of the ACM Multimedia*, 2010 International Conference

[5] Chris Cannam, Michael O. Jewell, Christophe Rhodes, Mark Sandler, and Mark d'Inverno: "Linked Data and You: Bringing music research software into the Semantic Web", *Journal of New Music Research*, Volume 39, no. 4, pp. 313–325, 2010

[6] Rasmus Berg Palm: "Prediction as a candidate for learning deep hierarchical models of data", *Technical University of Denmark, DTU Informatics*, 2012

[7] Yvonne Desportes and Alain Bernaud: "Manuel pratique pour l'approche des styles, de Bach Ravel", *Conservatoire National Suprieur de Paris*, 1979