A Multi-Layer Modeling Approach to Music Genre Classification

Renan V. Novas
Gabriel S. Vicente [*]

Abstract

Listening to music and understanding its patterns and structure is a fairly easy task for humans beings, even for listeners without formal musical education. However, building computational models to mimic those processes has proven a remarkably resilient problem. The algorithms and data associated to the Music Information Retrieval (MIR) field are both complex.

The scientific endeavour is motivated by a large class of challenging demands in the media business that require efficient and robust audio classification. Application scenarios include audio streaming services, such as Spotify and Pandora, automatic media monitoring, content-based search in multimedia databases, improvements on recommendation and filtering systems and last, but not least, purely artistic explorations.

1. Motivation

Musical genre is notoriously subjective concept and its classification has been a standard problem in the Musical Information Retrieval (MIR) research. It is a very active and multidisciplinary investigation field, comprising musicology, psychology, signal processing, artifial intelligence and machine learning. MIR applications can be divided typically in four groups.

- Instrument and musical structure recognition
- Recommendation and taste prediction engines
- Automatic music transcription and algorithmic composition
- Automatic categorization

We are here interested in the last. An extraordinary range of information is hidden inside of music waveforms, ranging from perceptual to auditory which inevitably makes large-scale applications challenging. There are a number of commercially successful online music services, such as Spotify, Pandora and Last.fm, but most of them are merely based on traditional text information retrieval. Many different features can be used for music classification, such as reference features including title and composer, content-based acoustic features including tonality, pitch, and beat, symbolic features extracted from the scores. Content-based music genre classification has been gaining importance and enjoying a growing amount of attention. Commonly used classifiers include Support Vector Machines (SVMs), Nearest-Neighbor (NN) classifiers, Gausian Mixture Models and Linear Discriminant Analysis (LDA).

2. Related Work

A long line of work addresses problems in the Music Information Retrieval. There are a growing interest in the scientic community and a vast diversity of new content-based models [1] [2]. The content-based acoustic features are classified into timbral texture features, rhythmic content features and pitch content features. Timbral features are mostly originated from traditional speech recognition techniques. They are usually calculated for every short-time frame of sound based on the Short Time Fourier Transform (STFT) and countless others audio descriptos, such as Mel-Frequency Cepstral Coefficients (MFCCs). More atypical descriptors, such as Daubechies Wavelet Coefficient Histograms (DWCH), have been used in experimentation [3].

3. Experiments and Discussion

We used several approaches to leverage different types of information about a song into a final classifier. First of all, we look in more detail to dataset.

3.1. Million Song Dataset

The Million Song Dataset (MSD) [4] is a freely-available collection of audio features and metadata for a million contemporary popular music tracks. The project emerged as a collaborative enterprise between The Echo Nest (now a subsidiary of Spotify) and Columbia University's Laboratory for the Recognition and Organization of Speech and Audio (LabROSA) to provide a dataset collection for evaluating audio-related research and to encourage algorithms

[*] **Contact**: ra116953@ime.unicamp.br

that scale to commercial sizes.

The dataset itself does not include any audio signal, only derived features. It constains nearly 300 GB of metadata and audio-descriptive data, available as an Amazon Public Dataset (AWS), or through a directly-downloadable subset consisting of 10,000 songs selected at random for a quick look.

- **Million Song Dataset**
  - 280 GB of data
  - 1,000,000 songs/files
  - 44,745 unique artists
  - 7,643 The Echo Nest tags
  - 2,321 MusicBrainz tags
  - 2,201,916 asymmetric similarity relationships
  - 515,576 dated tracks starting from 1922

Each audio track is associated to an exclusive Echo Nest song ID, to artist and song metadata and to numerical fields taken directly from the Echo Nest Analyze API. It is indeed a very extensive and scalable dataset.

However, the dataset involved in this present experiment is modest comparing to the MSD. Since the original collection does not explicitly provide genre information, we recreated a smaller dataset while using additional databases. The idea is to use artist tags that describe typical artist-genre associations to assign the information to each track. We used MusicBrainz [5] tags instead of hand-picking associations as they were applied by humans and usually very reliable. They also tend to be standardized, as MusicBrainz care for consistency. We applied the routine to a subset, yielding 16,00 different tracks, of which 90% were selected at random for training.

Table 1. List of genres groups and number of sample tracks

	Genre	Training	Test
A	dance and electronica	2,880	320
B	folk	2,880	320
C	jazz and blues	2,880	320
D	punk	2,880	320
E	soul and reggae	2,880	320
		14,400	1,600

Evidently, building such simplified dataset implies huge flaws. The main one is the unbalancedness of the data. In our case, we selected the same cardinality of songs to each group. A more subtle issue is the genre definition itself. Music genre is notoriously subjective concept. Indicating genre is far more complex than a trivial binary classification and often inconclusive and appropriate to ambiguity. We considered only artists that have been tagged consistenty. We could argue the label of a particular track, but they were still reasonable. It is a little extreme, but we wanted to avoid confusing artists, such as those that span more than one genre.

3.2. Audio features

As for audio features, timbre is represented as 12-dimensional vectors that are the principal components of Mel-frequency cepstral coefficients (MFCCs); they represent the power spectrum of sound, and are derived from Fourier analysis and further processing. MFCCs are very commonly used in speech recognition and music information retrieval systems. Also track level audio features such as loudness and tempo which captures the high level information of the audio. T empo is defined as number of beats per minute, or BPM and loudness is a real value number describes the general loudness of the song.

So use the simplest 30 audio features from The Echo Nest: loudness, tempo, time signature, key, mode, duration, 12 averages and 12 variances of timbre vectors.

3.3. Results and Analysis

We experimented at first a multi-class classification using 15 different support vector machines (SVMs) from the open-source Python implementation of the `Scikit-learn` project [6]. More detailed information is avaiable in the documentation. We set up kernel parameter to rbf, C to 1 and gamma to 0. The following code matrix served as input. *One-vs-One* correponds to 1-10 models and *One-vs-Rest* to 11-15. For those models, we mutiplied C by weights inversely proportional to the frequency of each SVM classes, in order to perform some balancing.

Table 2. Code Matrix

SVM	A	B	C	D	E
1	1	-1	0	0	0
2	1	0	-1	0	0
3	1	0	0	-1	0
4	1	0	0	0	-1
5	0	1	-1	0	0
6	0	1	0	-1	0
7	0	1	0	0	-1
8	0	0	1	-1	0
9	0	0	1	0	-1
10	0	0	0	1	-1
11	1	-1	-1	-1	-1
12	-1	1	-1	-1	-1
13	-1	-1	1	-1	-1
14	-1	-1	-1	1	-1
15	-1	-1	-1	-1	1

For each model, we considered 80% of random samples for training and 20% for a *leave-one-out* validation. All informations, in training, validation and test, were normalized using the mean values and stardard deviations of corresponding model.

Table 3. Scores

	Normalized Accuracies	
SVM	Training	Validation
1	92.5%	87.2%
2	88.4%	83.5%
3	91.2%	86.9%
4	93.7%	90.7%
5	89.8%	83.9%
6	87.6%	83.2%
7	90.6%	87.2%
8	90.1%	86.3%
9	93.5%	90.4%
10	91.9%	86.9%
11	86.7%	84.2%
12	88.1%	84.7%
13	87.7%	85.0%
14	84.3%	80.9%
15	91.8%	88.5%

3.4. A Multi-Layer Approach

Our first innocent attack has given us 15 differents classifiers. We offered them our initial dataframes, expecting predictions we may now attach as features to a more complex 2-layer model. We devised a *Extremely Randomized Trees* [7] approach, also implemented in `Scikit-learn` packages. We employed the *bagging* algorithm for constructing forest trees and *out-of-the-bag* for scoring success rate. Futhermore, in order to optimize the forest constructing parameters, we performed grid search. The best setting was 90 trees, 6 features at maximum and 3 samples for leaf at minimum.

Table 4. Confusion Matrix

	Predicted Genre					
	A	B	C	D	E	
A	224	21	24	17	34	320
B	12	224	39	15	30	320
C	27	42	223	9	19	320
D	24	15	7	240	34	320
E	39	36	16	11	218	320
	326	338	309	292	335	

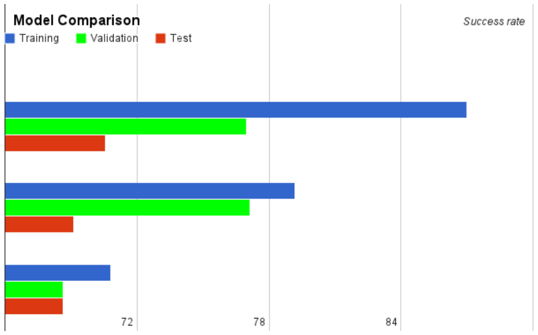
The average impact of SVM-related features was approximately 0.8%, with at most 7.46% of contribution. In fact, the original 30 features remained the most relevant. The success rate of 86.99% was attained in training and 76.9% in an *out-of-the-bag* scenario. In test, we accomplished 70.56% rate of successfully categorized songs. We achieved 70.56% of sucessfully categorizing tracks in the test dataset. Comparatively, more usual random forest models yields 69%-70%.

3.5. Alternative Solutions

In our experiments, we extended the test to several initial procedures. Using exclusively the SVM-related features, our best results were a single *random forest* [8] with 150 trees and 150 at maximum leaf nodes. The success rate reached 79.19% in training, 77.15% *out-of-the-bag* and 69.13% in test. *One-vs-One* models contributed with nearly 4%, while *One-vs-Rest* models were more important, holding 10%.

Using the original 30 features, the best result was again a *random forest* with 450 tress. Obtaining success on 70.79% of the training cases, 68.64% *out-of-the-bag* and 68.64% in test.

Comparing the three models in order of appearance:



Exploring further, multi-class classifiers such as ECOC, *One-vs-One*, *One-vs-Rest* without proper tuning do not yield impacting results, ranging from 35% to 40% success rate in test.

4. Conclusions and Future Work

In this project, we proposed a multi-layer modeling approach to music genre classification. The *Extremely Randomized Tree* model showed the best results, using SVM classfiers for feature extraction. We accomplished 70.56% of success rate in test. Looking at the confusion matrix, it is evident we avoided bias. Since genre tags were associated to artists in our experiment, musical compositions considerably different from the artist's usual characteristics and patters might explain some of the confusion. In the future, it would be interesting analysing cases in which the final classification is a more valid answer than our target valu Considering hypothetically the similiary matrices indicate stable results, it would possible to imply that some tracks are wrongly classified because its artist cannot be exclusively assigned to a particular genre. In the future, it would be convenient to analyse overlapping and ambiguity.

References

[1] Yajie Hu and Mitsunori Ogihara. Genre classification for million song dataset using confidence-based classifiers combination. In William R. Hersh, Jamie Callan, Yoelle Maarek, and Mark Sanderson, editors, *SIGIR*, pages 1083–1084. ACM, 2012.

[2] Ahmed Bou-rabee, Keegan Go, and Karanveer Mohan. Classifying the subjective: Determining genre of music from lyrics, 2012.

[3] Tao Li, Mitsunori Ogihara, and Qi Li. A comparative study on content-based music genre classification. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 282–289, New York, NY, USA, 2003. ACM.

[4] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[5] MusicBrainz Community. *MusicBrainz - The Open Music Encyclopedia*, 2014 (accessed October 3, 2014).

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[7] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.

[8] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.