A Combination of Hand-Crafted and Hierarchical High-Level Learnt Feature Extraction for Music Genre Classification

Julien Martel[1], Toru Nakashika[2], Christophe Garcia[1], and Khalid Idrissi[1]

[1] Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR 5205, France
firstname.name@insa-lyon.fr
[2] Department of System Informatics, Kobe University, Japan
nakashika@me.cs.scitec.kobe-u.ac.jp

Abstract. In this paper, we propose a new approach for automatic music genre classification which relies on learning a feature hierarchy with a deep learning architecture over hand-crafted feature extracted from an audio signal. Unlike the state-of-the-art approaches, our scheme uses an unsupervised learning algorithm based on Deep Belief Networks (DBN) learnt on block-wise MFCC (that we treat as 2D images), followed by a supervised learning algorithm for fine-tuning the extracted features. Experiments performed on the GTZAN dataset show that the proposed scheme clearly outperforms the state-of-the-art approaches.

Keywords: music genre classification, high-level hierarchy feature extraction, deep learning, deep belief networks.

1 Introduction

In the last decade, automatic music genre classification has become more important as the digital entertainment industry developed. Now audio files are distributed over the world wide web and are available as digital content with auxiliary-data also called meta-data. In order to search proper music from huge databases, labels in meta-data have to be assigned to each piece beforehand. The point in using a music genre classification system, is to assign them automatically instead of spending lots of effort in manual annotation.

Feature extraction from an acoustic music signal is a significant step in automatic music genre classification. Most systems in the early years mainly relied on timbre features extracted from a windowed short signal, such as MFCC, STFT, LPC, Filterbank Coefficients and Autoregressive Model [1]. Other methods employed statistical models of the timbre features such as histograms, means, variances, etc. [2]. These approaches, however, extract the features frame-by-frame and do not capture any temporal information.

As mentioned in [3], spectral transition in short term is considered to be an important factor for musical genre classification as well as timbre features of the frame. Meanwhile, a block-wise approach, where a feature descriptor involves multiple frames of a few seconds, has been gathering more attention in recent years [4]. One example can be found in [5], where 5 different block-wise features are obtained from 10 or 25 frames and are used for genre classification. Costa *et al.* attempted to extract texture features from a spectrogram of a few seconds, inspired by works in image processing [6]. There, the system extracts 7 statistical texture features after calculating a Gray Level Co-occurrence Matrix (GLCM) [7] from each spectrogram, and classifies the music signals using SVMs.

Another notable block-wise method was proposed by Tom *et al.* [8], where they adopted Convolutional Neural Networks (ConvNets) for bi-dimensional feature extraction and genre classification. In their work, the ConvNets learn music patterns given a bi-dimensional MFCC map and corresponding genre labels in a supervised way.

Besides, other Deep Learning approaches were used as a way to build hierarchical representations. One of the most well-known Deep Learning models is the Deep Belief Network (DBN) [9], which stacks multiple Restricted Boltzmann Machine (RBM) layers hierarchically. Hamel and Eck [10] adopted the DBN to learn high level musical features. In [11], a Sparse Encoder Symmetric Machine (SESM) was proposed by Ranzato *et al.* as another extension of DBN attempting to produce better representations in terms of sparseness. Since such deep learning methods rely on unsupervised feature extraction, it is expected that the models automatically extract more tractable and better-separated features for the supervised classifier.

In this work, we propose and study a deep learning approach for musical genre classification. We fed block-wise "hand-crafted" (by opposition to "learnt") features of MFCCs that are respectively inspired from [6] and [8] into a deep architecture in which we learn a higher level feature hierarchy with an unsupervised learning algorithm and use a supervised learning algorithm for fine-tuning using a set of known labels of songs. Experimental results on the well-known 10 musical genre GTZAN database show the efficiency of our approach.

2 The Proposed Approach

The deep learning method that we propose rely on Deep Belief Networks. Our architecture is based on the following functional blocks:
Hand-crafted features extraction: from the raw audio-file, we extract several block-wise hand-crafted MFCC features;
High-level learnt features extraction: from these features, we extract a hierarchy of high-level learnt features with an unsupervised learning algorithm;
High-level learnt features fine-tuning: we fine-tune these high-level "learnt" features to adapt them to our classification task with a supervised learning algorithm;
Classification: from this fine-tuned high-level hierarchy of features, we train a classifier such that each of the blocks votes for a certain genre;
Voting scheme: we collect the votes in a given voting space (binary, probabilistic scoring etc.) and output a genre associated to the whole music.

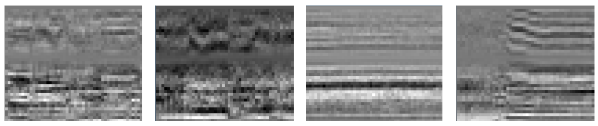

Fig. 1. Examples of block-wise MFCC extracted from the GTZAN database

2.1 Hand-Crafted Feature Extraction with Block-Wise MFCC

The idea behind using hand-crafted features on the "raw" music signal is to process it in a form that will be meaningful for the feature extractor we want to learn in a next step. These hand-crafted features, like Mel-Frequency Cepstral Coefficients (MFCCs) [12], capture a lot of engineering knowledge that has been developed over the years to extract relevant information in audio frames.

We therefore built our block-wise MFCC by computing the cepstrum coefficients over non-overlapping audio frames of 28 milliseconds. The quantized values on the mel-scale (in 40 bins) obtained for a MFCC are concatenated in a block with others computed in the next frame so that it constitutes a time-MFCC domain that we can regard as a bi-dimensional structure (an image). In our experiments, we chose to use 50 blocks so that the block-wise MFCC lasts 1.4 second. By using such block-wise MFCC, we aim at capturing temporal information (in the limit of the block size times the frame length) and timbral information thanks to the MFCC transformation [13]. Some examples of the MFCC blocks we obtain on audio files of the GTZAN database are shown in Figure 1.

2.2 Learnt High-Level Hierarchy Feature Extraction

One of the originalities of this work resides in the way we extract our block-wise features that can both capture timbre and temporal information. These two pieces of information are represented on 2D maps and can be regarded as "images" with a strong bi-dimensional structure. However, we do not directly classify these features with any supervised learning algorithm like in [14] because we know they extract relevant information but which is probably still "drown" and highly hidden in these features. Our idea is to "capture the regularities" in the music genre by first applying a powerful unsupervised learning algorithm which can statistically find a hierarchical structure in the input data. Moreover, one should see the design of these "hand-crafted" features as limited by the complexity human people can possibly put in it and by what they think is interpretable. This is legitimate to try to extract more from these features and combine the resulting objects in a way human would not be able to. Contrary to our method, other recent contributions directly worked on the audio signal by learning Deep Belief Net on raw audio files or on spectrum [10]. Our strategy that learns a hierarchy of features on the top of hand-crafted features performs better than these methods also probably thanks to the use of these hand-crafted features that consists in a first "rough-extraction" in the raw audio data.

In conventional learning methods for neural networks like back-propagation, problems arise when building deep architectures with many layers. If for certain applications a few layers can be sufficient to learn patterns from data, this is clearly not the case in audio especially in a genre classification task where the intra-class variability is very high. The fundamental hope in the proposed deep architecture is to better fit the underlying data with less hidden variables in each layer, more layers allowing to learn intricate "correlation" between these variables. Learning a good set of features, that is able to capture the main regularities of the model without capturing too much noise or "irregularities" make the parametrization of deep learning machines really challenging and subtle. However, they proved in many applications to be able to learn extremely interesting structures in features. In this work, we use an instance of deep learning strategies: the Restricted Boltzmann Machines as a building block for unsupervised learning stacked in Deep Belief Networks we then fine-tune in a supervised fashion.

The Restricted Boltzmann Machine. A Restricted Boltzmann Machine (RBM) is a type of Boltzmann Machine which is an instance of a probabilistic graphical model with interesting properties making inference tractable. The simplified connections result in a bidirectional bipartite graph composed of binary (that can be extended to real valued) stochastic units. On one side, a set of visible units $v = \{v_i\}_V$ receive the sensory input data (our block-wise MFFC), while, on the other side, hidden units $h = \{h_j\}_H$ can be regarded as holding an internal model representation. They are connected together by a set of weights $W = \{w_{ij}\}_{V \times H}$ under the assumption that the weights can be used symmetrically $w_{ij} = w_{ji}\ \forall (i, j) \in V \times H$. Pairwise energies can be defined over the so-formed network and result as an energy for the configuration of the different units: $E(\mathbf{v}, \mathbf{h}) = \mathbf{v}^T W \mathbf{h}$. This energy codes the correlation for any combination of two given binary units (in this case) to be "on" or "off" together. Then, a probability to be in a certain energy state (joint probability of the visible units) can be derived using the Boltzmann-Gibbs Distribution and as an analogy to statistical mechanics: $p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-\frac{1}{T} E(\mathbf{v}, \mathbf{h})}$ where Z stands for the partition function and T is an arbitrary temperature.

For RBMs, it is easy to analytically compute $p(h|v)$ and $p(v|h)$.

Two phases can be distinguished in the learning process: 1) a positive phase: when the visible data is clamped to the visible units and produces the activation of hidden units 2) a negative phase: with free visible units for which the hidden units produce a reconstructed visible data (an hallucination of the model). One possible learning procedure uses unsupervised learning and tries to minimize the reconstruction error between its own input and the reconstruction fantasized by the model. We use the so-called contrastive divergence: a simili negative-log likelihood learning procedure, for which the weights update is $\Delta w_{ij} = \epsilon (< v_i h_j >^+ - < v_i h_j >^-)$ where ϵ is the chosen learning rate and $< \cdot >$ is the expectation — the frequency of having v_i and h_j "on" together — computed respectively during the positive $(+)$ and negative $(-)$ phase.

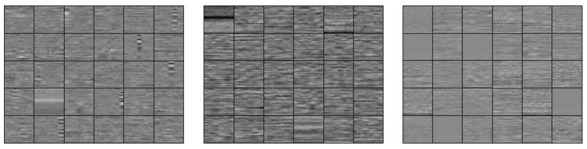

Fig. 2. From right to left : randomly selected features in each layer 1-3 of the hierarchy learnt on block-wise MFCC

Deep Architecture Used in the Experiment Set-up. We build a DBN with three RBM layers stacked on the top of each other and a single layer Perceptron. We clamp the block-wise MFCC on the 2000 units input layer and use 10 Perceptrons as outputs for classification. Concerning the hidden layer architecture we achieve dimensionality reduction by using two layers of 600 units in a row, with a high sparse penalty so that it helps to converge towards interpretable features with condensed information. The last hidden layer contains 2000 units to help the Perceptrons with linear separability. We used Gaussian visible units in the first RBM layer because we intend to use the real values provided by our block-wise MFCC. Gaussian units have shown to be good at learning real valued data and seem much better at properly modeling our problem than binary ones. A corollary is that it then takes much more time to learn such units. Using Gaussian units makes the learning signal theoretically unbounded; therefore the whole learning procedure might diverge very fast. That is why we use a low learning rate of 0.01 with a L_2 regularizer through weight decay to prevent an explosion in the weight values.

In Figure 2, we present some randomly selected features from the three layers of the Deep Belief Network learnt on our block-wise MFCC. To be visualized the feature intensities are shifted and scaled by a normalization process (centered on zero and scaled by their dynamic). Features in the second and third layers are back-projected in the first layer in order to be displayed.

2.3 High Level Learnt Feature Fine-Tuning and Voting Scheme

The fine-tuning step consists in using a supervised learning algorithm with the labels to help the high-level features we extracted previously to converge toward features specific to our classifier. The unsupervised learning procedure we run for RBM, namely "Contrastive-Divergence 1", does not make use of any label. This is a strength because it can be run over big-data for which labels have not been set. In this step, there is no need that the whole data we used previously to be labeled, as we might want to use only a fraction of it to fine-tune the classifier. As we target a 10-genre classification, we use a single layer of 10 Perceptrons on top of our high level feature hierarchy, each of them being set with a hyperbolic tangent activation function which outputs a confidence between -1 (not confident at all) and 1 (almost sure) for the associated audio genre.

2.4 Voting Scheme

The final step in our music-genre classification pipeline is to collect the confidence outputs of the Perceptrons for a single song given the different classification scores for each of the high-level features we extracted from each of the block-wise MFCC. We propose two very simple voting schemes.

One-shot Voting Strategy: In the first case, for each frame from which we extract our features, we perform a max operation over the classifiers (Perceptrons). The classifier with the most confident output for a certain genre casts a (+1) in a voting space consisting of the various genres, the other ones do not impact the vote. The genre which received the highest number of votes is declared the "winner" for the song.

Scoring Strategy: In the second scheme, each classifier votes according to its confidence in a genre. The confidence is scored by the output of the Perceptrons. Analogously the genre receiving the highest confidence is chosen to be the winner for the analyzed song. In our system, as the final Perceptrons are set with hyperbolic tangent activation functions, their scores range between -1 and 1 for each genre.

3 Experiments and Results

We conducted 10-musical-genre-classification experiments using the GTZAN dataset [1], which is widely used in this task. The dataset contains 100 songs for each of the following musical genres: Blues, Classical, Country, Disco, Hiphop, Jazz, Metal, Pop, Reggae and Rock (1000 songs in total). Some of them, —and we will experience it— are rather easily identifiable: it is usually the case of Reggae, Classical and Hard-rock. They usually do not give a lot of false negatives and have therefore a high recall rate. However, other genres are difficult to classify. For instance, the inter-class variability between rock and hard-rock is likely to be small against their intra-class variability and many false-positives of rock are recognized as hard-rock leading to a low precision. In GTZAN, each song is recorded during 30 seconds with a sampling rate of 22050 Hz using 16 bits quantization. We use a 10-fold cross validation for evaluating our method. For a fold, we select 90 songs from each genre for the training set (in total 900 songs) and the rest is used for the validation (100 songs). In order that all the songs are once used for training and validation, we proceed in such a manner 10 times and we average our results over the 10 trials. In table 1, we present the results of the classification after the fine-tuning step and after the voting scheme. The results clearly show that applying such a classification scheme provides very high classification rates: 96.0% of average recall rate on block-wise MFCC (before voting) and 99.8% on entire songs after voting with the score voting strategy. Table 2 clearly shows the superiority of the proposed approach when compared to the state-of-the-art approaches, with an important gap of 7.1% with the second best method.

Table 1. Per class statistics: Recall, Precision and F_1-measure on the block-wise MFCC after high-level features fine-tuning and Recall for the final voting scheme (2 strategies) using the fused high-level features from block-wise MFCC

	Block-wise MFCC			Fused features	
Genres	Recall	Precision	F_1-measure	One-shot Voting: Recall	Score Voting: Recall
Blues	0.97	0.97	0.98	1.00	1.00
Classical	0.98	0.97	0.98	1.00	1.00
Country	0.96	0.94	0.95	1.00	1.00
Disco	0.96	0.97	0.96	0.99	1.00
HipHop	0.95	0.96	0.96	0.98	0.98
Jazz	0.97	0.97	0.97	1.00	1.00
Metal	0.97	0.96	0.96	0.99	0.99
Pop	0.93	0.97	0.95	1.00	1.00
Reggae	0.96	0.95	0.95	1.00	1.00
Rock	0.95	0.94	0.95	1.00	0.99
Avg. (%)	96.0			99.7	99.8

Table 2. Overall results: recall rates (%) on the GTZAN music-genre classification problem, from [17] and completed with our results (in bold)

N^o	Classifier	Type of features	Recall
1	**Perceptrons**	**Learnt using DBN on MFCC**	**99.8**
2	CSC [15]	Many features	92.70
3	SRC [16]	Auditory cortical features	92
4	RBF-SVM [10]	Learnt using DBN on spectrum	84.3
5	Linear SVM [17]	Learnt using PSD on octaves	83.4
6	AdaBoost [4]	Many features	83
7	Linear SVM [17]	Learnt using PSD on frames	79.4
8	SVM [18]	Daubechies-Wavelets	78.5
9	Log. Reg. [19]	Spectral Covariance	77
10	LDA [14]	MFCC + other	71
11	Linear SVM [16]	Auditory cortical features	70
12	GMM [20]	MFCC + other	61

4 Conclusion

In this paper, we have proposed a new approach for musical genre classification which relies on learning a feature hierarchy over block-wise MFCC and outperforms the experiments that have been conducted until now. There are several reasons that may explain such an improvement of the state-of-the-art results. First, our strategy is based on MFCC that we tuned and customized to achieve temporal and timbral extraction. We then produce a set of different features which are of high-level and task specific thanks to the fine-tuning step. We also make use of sparsity because it is probably a privileged process to be able to learn such a number of features via the so called over-complete sparse representations. Concerning our classifier, we use a simple Single Layer Perceptron which could be "compared" to a linear-SVM used in most methods but that can refine the features previously found when fine-tuning with the back-propagation supervised learning. Unlike most approaches where each component may be trained (or fixed) "independently", our system is trained in a holistic way: each module influences the other ones during the different learning phases, leading to a powerful solution.

References

1. Tzanetakis, G.: Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing 10(5), 293–302 (2002)
2. Lidy, T., Rauber, A.: Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In: International Society for Music Information Retrieval Conference, pp. 34–41 (2005)
3. Tsuji, Y., Akahori, K., Nishikata, A.: The estimation of music genre using neural network and its educational use. In: International Conference on Computer-Assisted Instruction, pp. 158–162 (2000)
4. Bergstra, J., Kgl, B.: Aggregate features and adaboost for music classification. Machine Learning 2(65), 473–484 (2006)
5. Seyerlehner, K., Schedl, M., Pohle, T., Knees, P.: Using block-level features for genre classification, tag, classification and music similarity estimation. In: IMEX (2010)
6. Costa, Y., Oliveira, L., Koerich, A., Gouyon, F.: Music genre recognition using spectograms. In: WSSIP 2010, pp. 151–154 (2010)
7. Hua, B., Fu-long, M., Li-cheng, J.: Research on computation of glcm of image texture (2006)
8. Li, T.L., Chan, A., Chun, A.: Automatic musical pattern feature extraction using convolutional neural network. In: IMECS 2010 (2010)
9. Hinton, G.: To recognize shapes, first learn to generate images. Progress in Brain Research 165, 535–547 (2006)
10. Hamel, P., Eck, D.: Learning features from music audio with deep belief networks. In: International Society for Music Information Retrieval, pp. 339–344 (2010)
11. Ranzato, M., Boureau, Y.-L., Chopra, S., Lecun, Y.: A unified energy-based framework for unsupervised learning. Journal of Machine Learning Research 2, 371–379 (2007)
12. Bridle, J., Brown, M.: An experimental word recognition system, jsru report no 1003. Joint Speech Research Unit, Ruislip, England, Tech. Rep. (1974)
13. Li, T.L., Chan, A.: Genre classification and the invariance of mfcc features to key and tempo. In: International Conference on MultiMedia Modeling (2011)
14. Li, T.L., Tzanetakis, G.: Factors in automatic musical genre classification. In: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (2003)
15. Chang, K., Jang, J., Ilioupoulos, C.: Music genre classification via compressive sampling. In: International Society for Music Information Retrieval, pp. 387–392 (2010)
16. Panagakis, Y., Kotropoulos, C., Arce, G.: Music genre classification using locality preserving non-negative tensor factorization and sparse representations. In: International Society for Music Information Retrieval, pp. 249–254 (2009)
17. Henaff, M., Jarett, K., Kavukcuoglu, K., LeCun, Y.: Unsupervised learning of sparse features for scalable audio classification. In: International Society for Music Information Retrieval (2011)
18. Li, T.L., Ogihara, M., Li, Q.: A comparative study on content-based music genre classification. In: ACM SIGIR Conference on Research and Development in Information Retrieval (2003)
19. Bergstra, J., Mandel, M., Eck, D.: Scalable genre and tag prediction using spectral covariance. In: International Society for Music Information Retrieval (2010)
20. Smith, E., Lewicki, M.: Efficient auditory coding. Nature (2006)