MUSIC STYLE ANALYSIS USING THE RANDOM FOREST ALGORITHM

A. Gómez de Silva Garza and E. Herrera González

Computer Engineering Department, Instituto Tecnológico Autónomo de México (ITAM), Mexico City, Mexico

Abstract: This paper aims to discuss a method for autonomously analyzing a musical style based on the random forest learning algorithm. This algorithm needs to be shown both positive and negative examples of the concept one is trying to teach it. The algorithm uses the Hidden Markov Model (HMM) of each positive and negative piece of music to learn to distinguish the desired musical style from melodies that don't belong to it. The HMM is acquired from the coefficients that are generated by the Wavelet Transform of each piece of music. The output of the random forest algorithm codifies the solution space describing the desired style in an abstract and compact manner. This information can later be used for recognizing and/or generating melodies that fit within the analyzed style, a capability that can be of much use in computational models of design and computational creativity.

Keywords: style analysis, music style, markov model, wavelets, random forest

1. Introduction

When discussing an artist's style it is common for an outside observer (e.g., an art critic) to talk about the artist having followed certain rules or the artist's work containing certain patterns. These rules and patterns are generally used to describe not just one or two particular works of art, but rather an entire body of work—not necessarily the artist's entire body of work, but at least a subset of his/her historical output falling within a certain period or style. The rules that others attribute to an artist were not necessarily followed in a conscious manner by the artist, but they allow others to describe how they think the artist makes decisions. The patterns found in an artist's body of work are not necessarily obvious to everyone, and can depend on the amount of detailed analysis performed on the body of work by experts. It is these patterns that may be observed in an artist's body of work that we refer to in this paper as his/her "style." We are not interested here in trying to elucidate the rules—how the artist generates artwork that contains such patterns—only in detecting commonalities between the items in the body of work that is analyzed.

When the artist is a musician, many obvious and non-obvious patterns in the music are related to frequencies. Factors such as timbre (characteristic of each type of instrument), octaves used, notes used, execution speed, and many others—some perhaps more easy to detect than others, even if subconsciously, for non-expert listeners—can repeat themselves over and over within a musician's body of work. One approach to detecting such patterns is through the analysis of various exemplars that cover the musician's body of work (or the subset of interest of said body of work). This is the approach we have followed. In the rest of the paper we use the words exemplar, (musical) piece, and signal as synonyms to refer to one specific musical work of art produced by one artist.

One mathematical tool that can perform a detailed analysis of auditory frequencies is the Discrete Wavelet Transform (DWT). This transform takes a digital audio signal as input and uses a filter (a specific wavelet, where a wavelet is a brief oscillation that at some point crosses the horizontal or time axis, designed to "resonate" with a certain auditory property (Chui, 1992)) to produce a set of coefficients as output. These coefficients contain frequency-related information and together describe the input signal in a compact fashion and can be used to reproduce the original signal.

In the task of automating the analysis of a style through exemplars, it is important to use both positive and negative examples: exemplars that do and that don't pertain to the style of interest. We use the DWT to analyze each piece and then apply a Hidden Markov Model (HMM) analysis to the resulting coefficients. The HMM analysis extracts statistical information on frequency distributions that is implicit in the coefficients. This statistical information forms the HMM parameters associated with each piece.

After doing this for each piece, we use the random forest algorithm on the HMM parameters for the entire set of exemplars, both negative and positive. This produces a classification, where each original piece is classified as pertaining or not pertaining to the style being analyzed. This classification can be used by an autonomous music generationprogram to recognize whether its generated pieces fit within a style of interest or not, for instance if the program wants to imitate a style, without making exact copies of the original pieces that form that style.

The rest of this paper is organized as follows. Section 2 gives further details and explanations about our methodology for using a computer system to analyze a musical style. Section 3 discusses our domain of implementation and the values we selected for some of our methodology's parameters. After that in Section 4 we describe a series of experiments we performed and give the results of these experiments. In Section 5 we describe a possible application of our methodology and results to the automated generation of musical pieces that fit within a style. Finally we conclude with a discussion section where we review some related work and talk about some possible applications of our work.

2. Style Analysis Methodology

In this section we explain the methodology we followed for the analysis of a musical style. This methodology is based on three phases. The first phase is a frequency analysis of each musical piece using the DWT. The second phase is a statistical analysis of each musical piece using the HMM method. Once these two phases are complete for all the pieces, the third phase consists of a clustering algorithm that classifies the pieces, which in our work is the random forest algorithm. The next three sub-sections detail these three phases.

2.1 Frequency analysis using the discrete wavelet transform

Since we are interested in the analysis of frequency variations within a melody, it's convenient to work in the frequency domain. The Discrete Wavelet Transform (DWT) offers frequency-related information of the signal that interests us (but without discarding temporal information).

The DWT of a signal is obtained by passing the signal through a series of filters. Let x denote the input signal and g the impulse response of a wavelet (used as a low-pass filter). The DWT y is a vector of coefficients where each coefficient $y[n]$ is obtained as the convolution of x and g, where n is the index of the coefficient and k is the index of each discrete sample within the digital input signal:

$$y[n] = (x * g) = \sum_{k=-\infty}^{\infty} x[k]g[n-k] \tag{1}$$

In practice, instead of applying this equation, what is done is to use two wavelets, g and h, where g is a low-pass filter and h is a high-pass filter. This gives us a more detailed frequency analysis. By Nyquist's rule, this also permits us to discard half of the samples into which the input signal has been discretized (i.e., sub-sampling at half the frequency as it was originally discretized). This is done as follows:

$$y_{low}[n] = \sum_{k=-\infty}^{\infty} x[k]g[2n-k] \qquad (2)$$

$$y_{high}[n] = \sum_{k=-\infty}^{\infty} x[k]h[2n-k] \qquad (3)$$

The same analysis described by the two equations above can be applied recursively to the coefficients resulting from the first equation (the low-pass filter) in order to continue obtaining more and more detailed information, and this can be repeated as many times as necessary. The repetition can be continued until a limit is reached where discarding half of the (remaining) samples becomes redundant because there are too few samples of the input signal left. We could also repeat the process less than this theoretical limit, but some loss of precision in the output values would result. Figure 1 illustrates three levels of detail that can be achieved following the process described above.
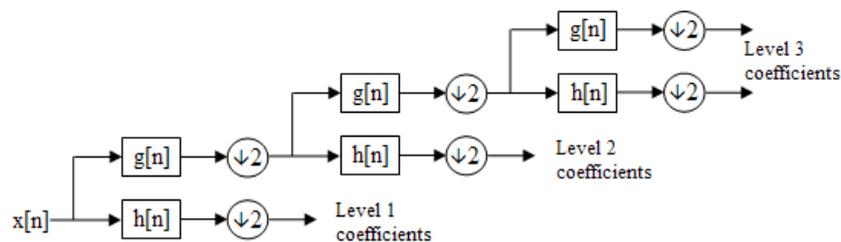
Figure 1. Three levels of decomposition of a signal using the DWT (taken from http://en.wikipedia.org/wiki/Discrete_wavelet_transform)

The result of this analysis is, at each level of decomposition, a set of approximation coefficients obtained by passing the signal through the low-pass filter and a set of detailed coefficients obtained by passing the signal through the high-pass filter. The high-pass filter in each level analyzes ever-decreasing frequencies (so in the end they're only high in relative terms) due to the sub-sampling. Both the approximation and the detailed coefficients contain information on the frequencies present in the input signal. At each level, all coefficients have dependencies between each other. Between levels, the values of the coefficients within each new level depend on the values of the coefficients of the previous level. These dependencies influence the structure of the Hidden Markov Model of the input signal (see below).

2.2 Statistical analysis using the hidden Markov model method

Hidden Markov Models (HMM) are widely used in signal processing applications like speech, handwriting, and gesture recognition. The main idea behind HMM is that a signal is assumed to be a Markov process (León-García, 2008) with unknown parameters, and the objective is to determine these hidden parameters from the ones that are observable (the input signals).

We assume that there is a pattern among the frequencies of a collection of signals associated with a particular musical style, so those parameters that identify a particular style are the ones that we will try to extract. The extracted parameters can be used in pattern recognition.

Because of the tree structure of the dependencies between the values of the coefficients obtained by the DWT, the HMM that describes this information, which is obtained by analyzing said dependencies, will also end up having a tree structure (Crouse, Nowak, & Baraniuk, 1998).

We can identify two types of variables in an HMM, observable and hidden. The observable variables for our purposes are directly the coefficients produced by the DWT. We use two hidden variables/states in our work, which are related to the amount of energy present in the signal. The values of the coefficients indirectly reflect this information. The two hidden states in the HMM

define the conditional probabilities associated with observing high-energy coefficients and low-energy DWT coefficients, respectively. The term "high-energy coefficient" is used here to refer to coefficients that are related to portions of the input signal with high information content (and vice versa for "low-energy coefficient").

The DWT gives the transition probabilities represented in the hidden states a tree-like structure because of the multiple levels of analysis involved, and this makes the model that results from the HMM analysis also have a tree structure. This resulting model consists of five parts: the observed variables (in the input signal) O, the hidden states S, the transition probabilities between states p and q, the probability of each hidden state being the initial state (one of them must be) r, and the conditional probabilities of each observed variable given which hidden state is the current state a. Figure 2 illustrates this information for a given level l of the tree structure (where L represents the total number of levels in the tree). The values of the probabilities are unknown at the beginning. We give them initial values that assume they follow a normal distribution before beginning the next phase, described below.
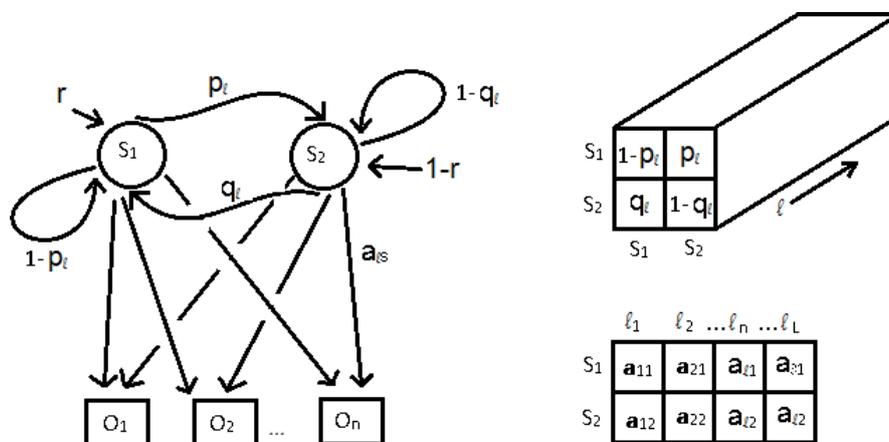
Figure 2. One level of description of the hidden Markov model of a signal (tree structure), including conditional probability tables for state transitions and possible observations

The final Hidden Markov Model obtained this way can be interpreted as a probabilistic representation of the input signal which can be used, when comparing with a similar representation of another signal, to measure how distant the two signals are from each other. The final values found for the probabilities p, q, r and a in the hidden Markov model are the "hidden parameters" we mentioned in the first paragraph of this section.

In our work we find the values of these probabilities by using the Expectation Maximization Algorithm (EMA). This is a general-purpose algorithm used to estimate a statistical model's parameter values (Dempster, Laird, & Rubin, 1977). EMA is an iterative algorithm which, for each coefficient obtained by the DWT Oi, uses the current probability values in the model to calculate the probability $P(Oi)$ of observing the coefficient and adjusts the values of the probabilities in the model to maximize the joint probability of all coefficients $P(\bar{O})$. This is repeated as many times as one indicates to the algorithm. The more iterations that are used, the more accurate the final probability values will be.

2.3 Classification of melodies using the random forest algorithm

Once we have an entire set of Hidden Markov Models with the final probability values, together they represent the entire subset of signals shown as exemplars of a particular musical style. Therefore, from the set we could extract a generic mathematical description of the entire style. In our work, we extract this generic mathematical description using the Random Forest learning Algorithm (RFA). RFA is a supervised learning algorithm for classification (Breiman, 2001).

The RFA creates a set of decision trees. This is the random forest from the algorithm's name. The size of the forest is specified by the user. The algorithm uses the Hidden Markov Models that the user gives it as a training set (TS). These models have to be labeled to indicate whether they represent an instance of the style that the algorithm is learning or not (i.e., to indicate which are positive and which are negative exemplars). Each of the decision trees in the forest randomly includes some of the probabilities from across several of the Hidden Markov Models in TS.

The forest that results from applying the RFA can be used to recognize whether a new signal belongs to the style of the positive training signals or not in the following fashion. The new signal's Hidden Markov Model must first be generated. This model is then presented to the forest. Each of the trees in the forest gives its opinion about whether the new signal is a positive or negative example of the style of interest. The final decision of the forest indicates the degree of fit of the new signal to the style, and is measured as the percentage of the trees in the forest that agreed that the new signal was a positive example of the style.

3. Domain and implementation

Francisco Tárrega was a Spanish composer until his death in 1909. He is considered to have laid the foundations for classical guitar composition in the 20th century. His most distinctive feature was playing classical compositions using the guitar instead of the more traditional piano. We decided to use Tárrega's as the style to process with our algorithm described above. To be able to do this we obtained 120 Tárrega pieces in MIDI format.

For each of these 120 pieces we extracted the first 13 seconds (approximately) to reduce processing time. Each of these 120 13-second segments was processed using the DWT to generate coefficients for 17 levels. Based on this, the Hidden Markov Model for each was found using the EMA for 30 iterations. We also found 110 pieces that were not Tárrega compositions (they were by Chopin, Tchaikovsky, Beethoven, Bach, Berlioz, Albéniz, Bizet, Haydn, Johann Strauss, and other classical composers) and followed the same process. We used these positive and negative sets to generate an appropriate random forest. We used 50 decision trees in the RFA.

4. Experimental results

We performed twenty experiments. The twenty differed only in the set of final decision trees produced by the RFA. In each case we split the 120 Tárrega exemplars into a training set of 100 and a testing set of 20, and the 110 non-Tárrega exemplars into a training set of 100 and a testing set of 10. In each experiment we followed our algorithm detailed above to produce a random forest using the training set, and we used it on the testing set, observing the random forest's accuracy of classification of this testing set. Table 1 shows the parameter values that were fixed across the twenty experiments and Table 2 shows the results for the twenty experiments. The accuracies shown in Table 2 represent the mean of the degrees of similarity of the different tested pieces to Tárrega's style.

Table 1. Invariant parameters across the twenty experiments

Parameter:	Size of positive training set	Size of negative training set	Size of positive testing set	Size of negative testing set
Value:	100	100	20	10

Table 2. Results of the twenty classification experiments

	Accuracy in identifying positive testing set	Accuracy in identifying negative testing set
Experiment 1:	100%	93.33%
Experiment 2:	100%	93.33%
Experiment 3:	100%	96.67%
Experiment 4:	95%	94.73%

Experiment 5:	100%	90%
Experiment 6:	100%	93.33%
Experiment 7:	100%	93.33%
Experiment 8:	100%	96.67%
Experiment 9:	100%	96.67%
Experiment 10:	100%	93.33%
Experiment 11:	100%	93.33%
Experiment 12:	100%	96.67%
Experiment 13:	100%	93.33%
Experiment 14:	100%	93.33%
Experiment 15:	100%	93.33%
Experiment 16:	100%	93.33%
Experiment 17:	100%	93.33%
Experiment 18:	95%	94.73%
Experiment 19:	100%	93.33%
Experiment 20:	100%	93.33%

From the data in Table 2, the mean accuracy of classification across the twenty experiments was 99.5% for the Tárrega exemplars and 93.97% for the non-Tárrega exemplars in the testing sets. This shows the robustness of our methodology, as even some of the random decisions involved in it (in particular, the choice of internal structure of the trees in the random forest) don't greatly affect the algorithm's accuracy when analyzing and classifying musical pieces that were not used in the training phase. The amount of false positives (non-Tárrega exemplars that our system incorrectly identified as having been composed by Tárrega) is 6.03%, a very good result for this type of classifier.

5. Style imitation

We plan to use an evolutionary algorithm (Mitchell, 1999) to compose new musical pieces. The evolutionary algorithm's fitness function could exploit the style description obtained as a result of the learning algorithm described above if we want to ensure that the evolutionary algorithm's compositions fit within the learned style.

Any evolutionary algorithm requires an initial population. To generate the initial population we plan to start from a simple musical scale (do, re, mi, etc.) of variable length (determined by the user). An alternative would be to start with some of the notes from one of the original pieces used to train the learning algorithm rather than a simple musical scale, and therefore provide more bias to the evolutionary algorithm. This sequence of notes is a crude individual in the evolutionary algorithm's population. The crude individual, whether it consists of a simple scale or notes that come from an original piece, will undergo a mutation process which modifies some of the properties of the notes at random and produces a new individual. The notes of the two individuals can be combined using the crossover operator common in evolutionary algorithms to produce two more new individuals.

These four individuals are considered the initial population (the first generation) of the evolutionary algorithm. They need to be evaluated to assign a fitness value to them. This fitness value represents the degree of fit of the individual to the learned style. This process should be repeated cyclically through several evolutionary generations until at least one individual's fitness value indicates a sufficiently close fit to the desired style. The population could grow indefinitely, but using elitism (keeping only the best individuals across generations in the evolutionary algorithm), and setting a maximum population size, ensures that this won't happen.

The evolutionary algorithm's progress is based on many random decisions (taking place in both the mutation and the crossover operators), so one might expect the search for an acceptable piece of new music to be slow. This is the point at which the potential bias that can be given to the evolutionary algorithm by starting out with an original piece of music rather than a simple scale can be helpful in guiding the search more rapidly towards an acceptable piece. In addition, the mutation and crossover

operators in an evolutionary algorithm are generally very fast, so even if the algorithm has to pass through many generations to find an acceptable solution, this doesn't necessarily indicate that an unacceptable amount of time will have passed.

This evolutionary method of generating new compositions automatically is similar to what can be observed in some human composers, who often refine their compositions over time by making slight modifications (mutations), and perhaps simple combinations with other compositions (crossover), eliminating what they don't like and keeping the characteristics that they do like until reaching a satisfactory composition.

6. Discussion

We have presented a methodology for analyzing a set of signals where the signals represent positive and negative exemplars of some characteristic of interest. In our case, the signals are digitalized pieces our music, and the characteristic of interest is belonging or not to a particular musical style. Our methodology consists of several phases. The first phase involves representing an auditory signal in the frequency domain to extract relevant information. The second phase consists of obtaining a statistical model, represented as a Hidden Markov Model, of the information in the frequency domain, using the expectation-maximization algorithm to approximate the values of the parameters that are used in the model. The third phase uses the random forest algorithm to identify patterns in, and produce a generic description of, an entire set of training exemplars (represented with their Hidden Markov Models). This generic description can then be used to analyze a new exemplar and measure a degree of fit to the patterns. This methodology is very similar to a previous work (Jafarpour, Polatkan, Brevdo, Hughes, Brasoveneau, & Daubechies, 2009), but in that work the signals represent visual information (specifically, paintings by van Gogh) and in ours they represent auditory information.

There is a description of similar work applied to musical signals (Chai & Vercoe, 2001). However, only Hidden Markov Models, without the random forest algorithm, were used, and their analysis was done without transforming the signal into the frequency domain (leaving it in the time domain). The accuracy results reported in (Chai & Vercoe, 2001) go from 54% to 77%, much poorer than the accuracy we achieved. Our use of a classification algorithm, and the fact that we used wavelets which contain both frequency- and time-related information about input signals, allowed us to achieve higher accuracy results.

Another project in which signals were categorized for the purposes of identifying style is (Jupp & Gero, 2006), though the signals in that case were visual, and the categorization was done using a Self-Organizing Map (SOM), also known as a Kohonen Map (Haykin, 2008). An interesting feature of the work in (Jupp & Gero, 2006) is that SOM's are unsupervised learning algorithms, so separating the training exemplars into positive and negative sets, as in our work, is not necessary.

The methodology we have described in this paper has several potential applications. One direct application is that it could be used in is the detection of forgeries. But now that we achieved high-accuracy recognition/classification of musical styles, another possibility is to automate the process of composing new music that fits within a given style, i.e., create "forgeries." We have included in the paper a description of how this could be done using an evolutionary algorithm. We are also thinking of using our methodology to analyze sets of cover versions of a particular song, and seeing whether our system can find the original author amongst them (having trained the system with several positive exemplars of that musician and several negative exemplars of the musicians that released the cover versions).

While these ideas for further research mention music explicitly, we assume that any type of signal could be processed successfully in the same way. Indeed, as mentioned in the previous paragraphs, other researchers have worked on similar projects and ideas but have applied them to visual signals/images rather than music. An extension of this, which we also believe to be perfectly feasible, would be to process not just artistic creations, but other types of designs (e.g., engineered artifacts) in the same way.

Acknowledgements

References

Breiman, L (2001). Random Forests. *Machine Learning* 45(1): 5-32.

Chai W, & Vercoe B (2001). Folk Music Classification Using Hidden Markov Models. *Proceedings of the International Conference on Artificial Intelligence.*

Chui CK (1992). *An Introduction to Wavelets Volume 1: Wavelet Analysis and its Applications*. San Diego, CA: Academic Press.

Crouse MS, Nowak RD, & Baraniuk RG (1998). Wavelet-Based Statistical Signal Processing Using Hidden Markov Models. *IEEE Transactions on Signal Processing*, Vol. 46, No. 4.

Dempster AP, Laird NM, & Rubin DB (1977). Maximum Likelihood from Incomplete Data Using the EM Algorithm. *Journal of the Royal Statistical Society Series B (Methodological)* 39(1): 1-38.

Haykin S (2008). *Neural Networks and Learning Machines*, 3rd edition. Upper Saddle River, NJ: Prentice Hall.

Jafarpour S, Polatkan G, Brevdo E, Hughes S, Brasoveneau A, & Daubechies I (2009). Stylistic Analysis of Paintings Using Wavelets and Machine Learning. *Proceedings of the Seventeenth European Signal Processing Conference (EUSIPCO '09)*, Glasgow, 1220-1224.

Jupp J, & Gero JS (2006). Visual Style: Qualitative and Context Dependent Categorisation. *Artificial Intelligence in Engineering, Design, Analysis and Manufacuring*, 20(3) 247-266.

León-García A (2008). *Probability, Statistics, and Random Processes for Electrical Engineering, 3rd edition*. Upper Saddle River, NJ: Prentice Hall.

Mitchell, M (1998). *An Introduction to Genetic Algorithms (Complex Adaptive Systems Series)*. Cambridge, MA: MIT Press.