Topological Representation of Digitized Signal Features of Music for Automatic Playlists Generation

Héctor Hidalgo*, Héctor Allende* and Rodrigo Salas+

*Departamento de Informática, Universidad Técnica Federico Santa María, UTFSM, Valparaíso, Chile. e-mail: hhidalgo@inf.utfsm.cl; hallende@inf.utfsm.cl

+ Departamento de Ingeniería Biomédica, Facultad de Ciencias, Universidad de Valparaíso, Chile. e-mail: rodrigo.salas@uv.cl

Keywords: Data Mining; Clustering; Music Information Retrieval; Feature Extraction of Music Files; Automatic Playlist Generation.

Abstract

In this work we propose an automatic method to generate playlists of music based on a topological representation of many signal features obtained from thousands of digitized music files. The method consists in a pattern recognition system where features are extracted from the digitized music signal and they are clusterized by similarity with a modified version of the Self Organizing Maps.

Simulations results show that it is possible to automatically generate playlists of similar music with this approach.

1 Introduction

Since the creation of digital audio formats, huge amounts of audio files began to be stored in commercial system, on personal computers and even in mobile devices. These large quantities of files need to be more than just simply stored and organized, but they should also be automatically processed for tasks such as classification, clustering by style similarity, or processing new music from among millions of musical groups worldwide.

Nowadays, the music playlist generation problem is dealt with a pattern recognition system that has several stages: feature extraction, feature selection, automatic classification, and finally, music playlist generation. Regarding to the feature extraction process there are several proposals in the literature; however the work of Tzanetakis et al. [10] stands out because it studied the exhaustive extraction of features from digital files. Yoshii et al. [14] proposes an analysis of a mixture of features, spectral features in addition to user tags in online systems; and Wang et al. [12] discusses the dilemma of what kind of features are best for artistic style clustering. On the other hand, approaches for feature selection are also diverse. For example, while Mierswa et al.[7] proposes the assessment of fitness of the extracted features, Ellis [1] gives a priori preference on the type of features that will be used (in this case chroma features). For the classification task the following problems have been studied: the work of Tzanetakis et al. [11] identifies artists, music, or genre, and the work of Klapuri [6] studies the visualization of large music collections and their transcription. Finally, for music playlist generation the following works stand out: Gulik et al. [3] proposes a technique for generating visual playlists with small interventions of a user, Flexer et al. [2] introduces a technique that does not use metadata but is based on features like MFCC, and the approach presented by Hu et al. [5] uses standard information retrieval techniques.

2 Methodology

Figure 1 schematically depict the proposed automatic method to generate playlists of music based on a topological representation of many signal features obtained from thousands of digitized music files.

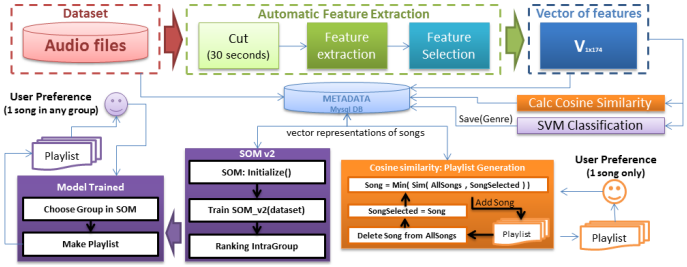

Figure 1. Methodology of the overall process for generating playlists.

In the following sections we will explain each stage of the processing method.

2.1 Feature extraction

From the digitized music files, several features are extracted with the methods described below.

- **Spectral Centroid:** is a measure of the spectral shape and corresponds to the balance point of the magnitude spectrum of the STFT.
- **Spectral Rollof:** is defined as the frequency at which the 85% of the magnitude of the distribution is concen-

trated, i.e. at the lowest frequencies. The Rollof is also a measure of spectral shape.

- **Spectral Flux:** is the 2-norm of the difference between the magnitude spectrum of the Short Fourier Transform (STFT) evaluated in two successive frames and it is a measure of spectral change.
- **Zero Crossings:** provides a measure of how noisy is a signal. It is the number of times the signal changes sign (or touches the zero value) and it is a good measure for estimating the pitch.
- **MFCC:** is a representation based on a cosine transform of a log power spectrum on a nonlinear mel scale of frequency.
- **LPCC:** are linear coefficients (LPC) represented in the cepstrum domain and provide estimates of the parameters of speech.
- **Chroma Features:** are a interesting representation of the audio, because the entire spectrum is projected into 12 parts representing the 12 different semitones (or chroma) of a musical octave (C, C#,D,D#,...,B).
- **Spectral Flatness Measure:** is a measure used to characterize an audio spectrum. A high measure of this feature might indicate the presence of white noise; on the other hand, a low measurement would indicate that the power spectrum is concentrated in a small number of bands.
- **Spectral Crest factor:** like the above feature it is related to the spectrum peaks. It is used to represent the relationship between the peak to the RMS value of a waveform measured in a specified time interval.
- **Line Spectral pairs (LSP):** are based on mathematical modeling of the vocal tract, which is conceived as a series of tubes of variable section where the sound propagates as a plane wave through the tubes from the glottis to the lips.

In summary, the features that were extracted in this work are: Spectral Centroid (1 feature), Spectral Rollof (1 feature), Spectral Flux (1 feature), Zero Crossings (1 feature), Peak Ratio Average (1 feature), Peak Ratio Minimum (1 feature), Mel Frecuency Cepstral Coefficients (MFCC) (13 features), Linear Prediction Cepstral Coefficients (LPCC) (12 features), Chroma Features (12 features), Spectral Flatness Measure (24 features), Spectral Crest factor (24 features), and Line Spectral pairs (LSP) (18 features).

The means and standard deviations of these 109 features were calculated over a "texture" window of 1 second consisting of 40 "analysis" windows of 20 milliseconds (512 samples at 22050 sampling rate) producing 436 new accumulated features which correspond to a vector representation of each music file [4]. The feature calculations are almost all based on the Short Time Fourier Transform (STFT) that can be efficiently calculated using the Fast Fourier Transform (FFT) algorithm. Each dataset was processed in order to obtain the above features, and subsequently by calculating averages and standard deviations, a N-dimensional vector for each MP3 file was obtained.

2.2 Feature Selection and Automatic Music Classification

Of all possible approaches to perform the selection phase, the Weston et al. [13] proposal was used because it corresponds to a wrapper method that can analyze the performance of a subset of features selected at each step, while it removes those less relevant features by using a backward algorithm.

For the classification task, the algorithm "sequential minimal optimization" for training a support vector classifier with a RBF kernel was used [8]. In this case, the multi-class problem was solved by using the pairwise coupling method.

Figure 2 shows the *"peaking phenomenon"* [9] that was observed by removing the less relevant features up to the most relevant ones. With the inspection of this figure,174 features were selected where they should be best ranked in at least two of the three datasets.

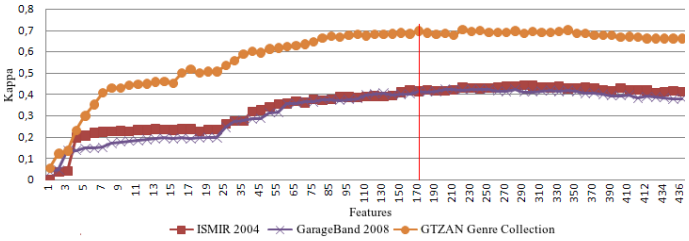

Figure 2. Variation of the Kappa measure vs number of features selected. 172 features were selected from this analysis. Datasets tested: ISMIR 2004; Garageband 2008; GTZAN Genre Collection which are available online.

2.3 Playlist generation

For the playlist generation, we propose to slightly modify the classical Self-organizing map (SOM) model to learn the topology of the feature space of digitized music, in order to "show and to produce" a music playlists. The modifications consists in...In what follows we will call this algorithm as *SOMfdm*, where *fdm* stands for *features of digitized music*.

Algorithm *SOMfdm*

```
1 Randomize the map's nodes weight vectors.
2 Find a BMU Best Match Unit). If BMU is
      already occupied, find another BMU in
      the neighborhood.
3 If all of nodes in the neighborhood are
      occupied, wide the radius and find the
      BMU in the extended neighborhood.
      Repeat this, until a BMU is found.
4 Save the BMU like a centroid found in the
      step 2 and save like a position the new
       BMU found in step 2 or step 3.
```

```
5 Estimate the BMU's neighbors found in step
      4 according to some criterion. In this
      case, according to what is shown in
      Figure \ref{SOMfigure}.
6 Alter the weights of all the neighbors,
      according to the exponential decay
      function that allows closer the
      similarity between each node in the
      neighborhood and the input vector.
7 Repeat from step 2 for N iterations.
```

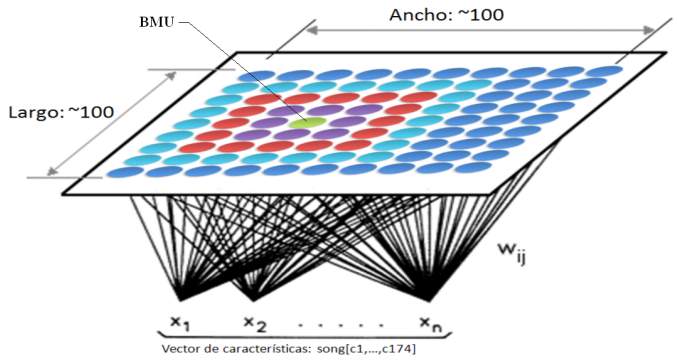

Figure 3. Procedure for searching for the position of a new audio file on the grid of neurons. If the position of the BMU is already being occupied, looking for a new BMU in the neighborhood according to this procedure

3 Results

For the experiments we have used the following public available datasets: *GarageBand 2008*, *ISMIR 2004* and *GTZAN Genre Collection*.

In Figures 4, 5 and 6 we can observe similar results in the output layer to the traditional SOM network for datasets ISMIR2004 , GarageBand and GTZAN respectively, whereas each dataset has about ten classes, this approach does not offer an intuitive classification or similarity measure for itself.

In figures 7, 8 and 9, we note that training *SOMfdm* in the output layer generates a good grouping of Mp3 in contrast to the provided by traditional SOM network (small number of groups and lots of songs for each music group). On the other hand in figures 10, 11 and 12 we note that the *SOMfdm* generates nonclustered songs and seemingly random distribution of neurons in the grid.

In figures 13, 14 and 15 we note that the quantity of groups is also small in contrast to the result of the training of a traditional SOM network, but with a greater quantity of groups than those obtained without processing the feature vectors before the training step.

Finally, figures 16, 17 and 18 we note that when you add attributes of order (before training) to the estimated class of the MP3, we get similar numbers of islands of data in the output layer (like the previous method), but with a larger grouping of styles, which facilitates the playlists generation.

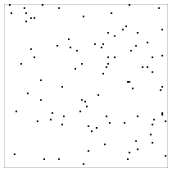

Figure 4. ISMIR 2004

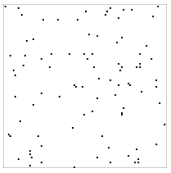

Figure 5. GarageBand

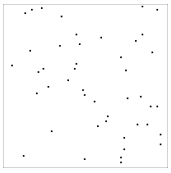

Figure 6. GTZAN

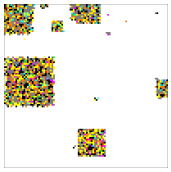

Figure 7. ISMIR in *SOMfdm* without preprocessing features

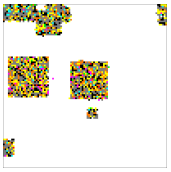

Figure 8. GarageBand in *SOMfdm* without preprocessing features

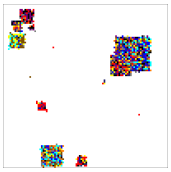

Figure 9. GTZAN in *SOMfdm* without preprocessing features

The following are two fragments of playlists generated by the method of cosine distance and neural network *SOMfdm*. For to see additional results and experiments please refer [4]

- Playlist 1 was obtained with cosine distance. In a preliminary phase, the playlist appears to be consistently similar, however, considering the initial point (user choice), which in this case is a *ROCK SONG*, the next song corresponds to a style *METAL SONG*, but then the approach recommends a style that could not be that the user wanted.(Please refer the work of H. Hidalgo for further details [4])

- Playlist 2 shows a fragment of the playlist generated on the basis of the normalization of feature vectors and the estimated class produced by SVM. Both are preliminary processes to the training of the neural network *SOMfdm*. It can be seen that the transitions are generally more acceptable as they are mixed well aspects of similarity based on musical genre and BPM.

Fragment of the Playlist 1
Using cosine similarity

N	Dataset Class	Mp3	BPM
1	rock	31.mp3	117
2	metal	18.mp3	113
3	jazz	02.mp3	184
4	jazz	46.mp3	156
5	jazz	35.mp3	141
6	hiphop	13.mp3	135
7	jazz	40.mp3	145
8	blues	24.mp3	116

Fragment of the Playlist 2 with the application of *SOMfdm*
the input was normalized, and SVM was used for classification

N	Centroid	Neuron	BPM	Class	Dataset Class
1	(3,4)	(3,4)	107	blues	blues.59.mp3
2	(3,4)	(4,5)	172	blues	blues.40.mp3
3	(3,4)	(2,3)	124	country	country.01.mp3
4	(3,4)	(2,5)	132	disco	disco.75.mp3
5	(3,4)	(2,4)	130	disco	disco.34.mp3
6	(3,4)	(3,3)	95	disco	hiphop.10.mp3
7	(3,4)	(4,4)	172	disco	country.09.mp3
8	(3,4)	(4,3)	114	hiphop	hiphop.21.mp3

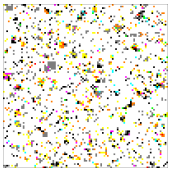

Figure 10. ISMIR in *SOMfdm* with standardized features

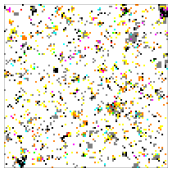

Figure 11. GarageBand in *SOMfdm* with standardized features

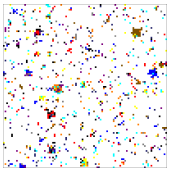

Figure 12. GTZAN in *SOMfdm* with standardized features

Figure 16. ISMIR in *SOMfdm* with features in [0,1] + SVM classification

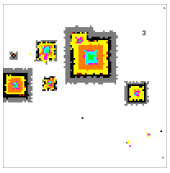

Figure 17. GarageBand in *SOMfdm* with features in [0,1] + SVM classification

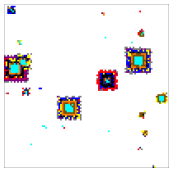

Figure 18. GTZAN in *SOMfdm* with features in [0,1] + SVM classification

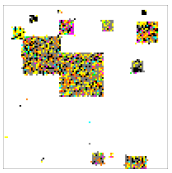

Figure 13. ISMIR in *SOMfdm* with features in [0,1]

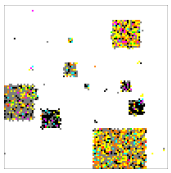

Figure 14. GarageBand in *SOMfdm* with features in [0,1]

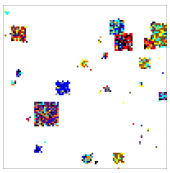

Figure 15. GTZAN in *SOMfdm* with features in [0,1]

First we should note that the cosine similarity method proposed in this work, can not provide appropriate recommendations. However, this does not imply that the similarity measure is not very suitable and requires further study.

The results obtained with a traditional SOM network, improve the proposal given by the cosine distance method, however, requires a rank element to generate playlists. The playlists generated by the self-organizing map *SOMfdm* were better than those generated by traditional SOM neural network but better results were obtained if the features are normalized. The results are improved if the approach uses the estimated class by SVM model and they are included as an element of ranking at the beginning of the process that generates playlists using *SOMfdm*.

4 Concluding Remarks

This paper shows that it is possible to automatically generate playlists of similar music with this approach. It is noted further that, although the vector representation is summarized in contrast to the complete data of each audio signal, there is quantitative information in this representation that allow automatic learning tasks.

Finally, we conclude that the the playlists generated can be improved if the model is extended by adding new features or parallel techniques, opening the possibility to include in addition to the quantitative information contained in this paper, qualitative information which may be obtained from the users when they interact with the playlists generated by the model.

Future work involves: research what section of an audio signal is most suitable to process, study the changes occurring in the performance of machine learning algorithms to vary the sampling rates, bitrates, etc., continue the study of features that should be included in the machine learning algorithms, and research other methods of ranking for generating music playlists.

Acknowledgements

Part of this research has been funded by the research grant FONDECYT 1110854 and DIPUV 37/2008 Project.

References

[1] D. Ellis. Classifying music audio with timbral and chroma features. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR-07)*, pages 339–340, 2007.

[2] A. Flexer, D. Schnitzer, M. Gasser, and G. Widmer. Playlist generation using start and end songs. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR-09) - Session 2a - Music Recommendation and Organization*, pages 173–178, 2008.

[3] R. Gulik and F. Vignoli. Visual playlist generation on the artist map. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR2005)*, pages 520–523, 2005.

[4] H. Hidalgo. Representación topológica de las características de la señal digitalizada de la música para la generación automática de listas de reproducción. Master's thesis, Departamento de Informática. Universidad Técnica Federico Santa María, Chile, 2012.

[5] X. Hu and J. Liu. Evaluation of music information retrieval: Towards a user-centered approach. In *Proceedings of the 4th Workshop on Human-Computer Interaction and Information Retrieval (HCIR2010)*, 2010.

[6] A. Klapuri and M. Davy, editors. *Signal Processing Methods for Music Transcription*. Springer, New York, 2006.

[7] I. Mierswa and K. Morik. Automatic feature extraction for classifying audio data. *Machine Learnearning*, 58(2-3):127–149, February 2005.

[8] J.C. Platt. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Technical Report MSR-TR-98-14, Microsoft, Redmond, Washington, 1998.

[9] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition*. Elsevier, 2009.

[10] G. Tzanetakis and P. Cook. MARSYAS: a framework for audio analysis. *Orginised Sound*, 4(3):169–175, 1999.

[11] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293 – 302, jul 2002.

[12] D. Wang, T. Li, and M. Ogihara. Are tags better than audio features? The effect of join use of tags and audio content features for artistic style clustering. In *11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, pages 57–62, 2010.

[13] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs. In *Advances in Neural Information Processing Systems 13*, pages 668–674. MIT Press, 2000.

[14] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H.G. Okuno. An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):435 –447, 2008.