Self-Organizing Maps for Content-Based Music Clustering

Markus Frühwirth, Andreas Rauber
Department of Software Technology, Vienna University of Technology
Favoritenstr. 9 - 11 / 188, A–1040 Wien, Austria

Abstract

With the increasing amount of music available electronically, methods for organizing these collections to allow intuitive browsing and orientation gain importance. Due to the large amounts of data involved, conventional approaches to organize music by genre or musical style are only of limited applicability, commonly relying on textual descriptions and manual classification. This makes it a particularly challenging application arena for neural networks capable of handling very high-dimensional input spaces and the noisy patterns associated with musical data.

In this paper we present a system based on the *Self-Organizing Map* which automatically organizes a collection of music files according to their musical genre and sound characteristics. Frequency spectra are used to extract feature vectors describing sound and melody characteristics. A two-stage clustering procedure first groups music segments according to their similarity, followed by a clustering of compositions according to the segment similarities. As a result, pieces of music with similar sound characteristics are found in neighboring regions of the resulting map, thus offering a very intuitive interface to unknown music collections.

1 Introduction

The wider availability of cheaper high-tech music recording equipment resulted in a tremendous rise of music data available electronically. Apart from the much-criticized pirated copies of copyrighted labels, many independent composers and smaller bands make their recordings publicly available for little or no fees at all via public domain music libraries such as AudioGalaxy.com. Contrary to well-known composers and bands, where users commonly know the style and characteristics of their favorite stars, finding pieces of music to suit ones taste is rather difficult in this public domain setting. In order to help users in finding their way through the piles of publicly available pieces of music from lesser-known groups, music portals try to provide a manual classification of the titles they offer. This way of organizing and presenting music closely mirrors the way music is presented in conventional stores, where we also frequently find CDs to be organized first by musical genres, within which an alphabetical organization is followed. Yet, providing such a manual classification becomes increasingly difficult with the amount of music submitted every day increasing. Furthermore, the resulting classification into any musical genre hierarchy is highly subjective. These effects are even worse when the classification is performed by several persons, such as by the performing artists themselves.

In order to cope with this challenge, methods for automatically organizing music by genre gain importance. Due to the difficulties of analyzing the content of music

itself, most approaches reverted to text-based analysis of pieces of music, relying on title and author information, metadata description, or the lyrics of songs for automatic classification. These features form the core of the search facilities of the MPEG7 standard currently under development [7]. Similar to manual classification, these approaches to finding and organizing music rely heavily on manually created descriptions. A different line of research is constituted by content-based music analysis, trying to organize and locate pieces of music based on the similarity of melodies. The digital music library [4, 1] extracts melody-information from a hummed query and matches it against a database of musical tunes for which the actual scores are available. Similar approaches are reported in [6], using the scores provided by MIDI-files to index and retrieve musical documents, and in [3], focusing on beat detection.

Yet, for the majority of music documents available today, such as the prominent MP3 files, no musical scores are provided. What we would thus like to have is a way to provide content-based organization and retrieval of musical documents based on the actual sound rather than on score transcripts. However, with the huge amounts of data used for describing sound information as well as the inherent noise in musical sound representation, conventional retrieval techniques are of only limited use. This makes it a challenging arena for neural networks, which are particularly suited for generalizing from noisy data and for extracting key features from large datasets.

In this paper we propose a content-based clustering of musical documents based on the actual sound. Rather than trying to extract precise scores, frequency spectra are used to describe the characteristics of a specific piece of music. We then use the *Self-Organizing Map* (*SOM*) [5], a popular unsupervised neural network, to automatically cluster pieces of music according to their similarity. After the unsupervised training process, similar pieces of music are found in neighboring areas on the two-dimensional map display. This allows a user to easily orient herself within an unknown music collection, by finding, say, classical music in the upper left corner of the map, whereas disco-style music may be found in a different region. Selecting a cluster of music according to ones current preferences, rather than having to specify a list of songs based on textual descriptions provides a more intuitive and direct access to music libraries. These concepts have successfully been applied to text clustering [2, 8].

The remainder of this paper is structured as follows: Section 2 presents the architecture of our system, detailing feature extraction, vector creation and music clustering using the *Self-Organizing Map*. We then provide experimental results using a collection of MP3 files in Section 3 and finally some conclusions as well as an outlook on future work in Section 4.

2 Clustering of Music

Music comes in a variety of file formats such as MP3, WAV, AU, etc., all of which basically store the sound information in the form of pulse code modulation (PCM) using a very high sampling rate of 44.1 KHz. The analog sound signal is thus represented by 44.100 16 bit integer numbers per second, which are interpreted by media players to reproduce the sound signal. To be able to compute similarity

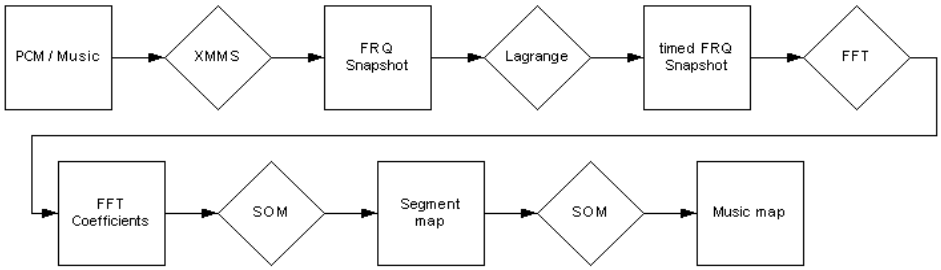

Figure 1: System Architecture: feature extraction, conversion, SOM training

scores between musical tunes, a feature vector representation of the various pieces of music needs to be created, which can further be analyzed by the *SOM*. Figure 1 provides an overview of the system architecture.

Starting with any popular music file format, most media players, such as the public domain X Multimedia System (XMMS) are capable of splitting this data stream into several frequency bands. Using the XMMS the signal is split into 256 frequency bands, with approximately one sample value every 20 to 25 ms each. Since not all frequency bands are necessary for evaluating sound similarity and in order to reduce the amount of data to be processed, a subset of 17 frequency bands (i.e. every 15th frequency band) is selected for further analysis, covering the whole spectrum available. In order to capture musical variations of a tune, the music stream is split into sections of 5 seconds length, which are further treated as the single musical entities to be analyzed. While basically all 5-second sequences could be used for further analysis, or even overlapping segments might be chosen, experimental results have shown that appropriate clustering results can be obtained by the *SOM* using only a subset of all available segments. Especially segments at the beginning as well as at the end of a specific piece of music can be eliminated to ignore fade-in and fade-out effects. Specifically, our results show that choosing every second to third segment, i.e. a 5-second interval every 10 to 15 seconds, provides sufficient quality of data analysis.

The intervals between the frequency snapshots provided by the player varies with the system load and can thus not be guaranteed to occur at specified time intervals. We thus have a set of amplitude / timestamp values about every 20 to 25 ms in each of the 17 selected frequency bands. In order to obtain equi-distant data points, a Lagrange interpolation is performed on these values as provided in Expression 1, where $f(x_i)$ represents the amplitude of the sample point at time stamp x_i the data points for up to $n+1$ sample points.

$$P_n(x) = \sum_{i=0}^{n} \left(\prod_{j=0, j \neq i}^{n} \frac{x - x_j}{x_i - x_j} \right) f(x_i) \tag{1}$$

As a result of this transformation we now have equi-distant data samples in each frequency band.The resulting function can be approximated by a linear combination of sinus and cosines waves with different frequencies. We can thus obtain a closed representation for each frequency band by performing a Fast Fourier Transformation

(FFT), resulting in a set of 256 coefficients for the respective sinus and cosines parts. Combining the 256 coefficients for the 17 frequency bands results in a 4352-dimensional vector representing a 5-seconds segment of music. These feature vectors are further used for training a *SOM*.

The *Self-Organizing Map* [5] is one of the most prominent artificial neural network models adhering to the unsupervised learning paradigm. It provides a mapping from a high-dimensional input space to a usually two-dimensional output space while preserving topological relations as faithfully as possible. Input signals $x \in \Re^n$ are presented to the map, consisting of a grid of units with n-dimensional weight vectors, in random order. An activation function based on some metric (e.g. the Euclidean Distance) is used to determine the winning unit (the 'winner'). In the next step the weight vector of the winner as well as the weight vectors of the neighboring units are modified following some learning rate in order to represent the presented input signal more closely. As a results, after the training process, similar input patterns are mapped onto neighboring units of the *Self-Organizing Map*. The feature vectors representing music segments can be thought of data points in a 4352-dimensional space, with similar pieces of music, i.e. segments exhibiting similar frequency spectra and thus similar FFT coefficients, being located close to each other. Using the *SOM* to cluster these feature vectors, we may expect similar music segments to be located close to each other in the resulting map display.

Using the resulting segment *SOM*, the various segments are scattered across the map according to their mutual similarity. This allows, for example, pieces of music touching on different musical genres, to be located in two or more different clusters, whereas rather homogeneous pieces of music are usually located within one rather confined cluster on the map. While this already provides a very intuitive interface to a musical collection, a second clustering may be built on top of the segment clustering to obtain a grouping of pieces of music according to their overall characteristics. To obtain such a clustering, we use the mapping of the segments representing a single piece of music to obtain an overall clustering. We thus create a feature vector representation for each piece of music using the location of its segments as descriptive attributes. Given an $x \times y$ *SOM* we create an $x \cdot y$ dimensional weight vector, where the attributes are the (coordinates of) the units of the segment *SOM*. Each vector attribute represents the number of segments of a particular piece of music mapped onto the respective unit in the *SOM*. For example, given a piece of music that has 3 segments mapped onto unit (0/0) in the upper right corner of the map, and 2 segments on the neighboring unit (1/0), the first two attributes of the song's feature vector are basically set to the according values $(3/2/\ldots)^T$, with subsequent norming to unit length to make up for length differences of songs. Training a second *SOM* using these feature vectors we obtain a clustering where each piece of music is mapped onto one single location on the resulting map, with similar pieces of music being mapped close to each other.

3 Experiments

For the following experiments we use a collection of 230 pieces of music, ranging from classical music, such as *Mozart's "Kleine Nachtmusik"*, via some hits from the

1960's such as *Cat Steven's "Father and Son"* or *Queen's "I want to break free"*, to modern titles, e.g. *Tom Jones' "Sexbomb"*.

These songs were segmented into 5-second-intervals, of which every second segment was used for further processing with a total of 17 frequency bands being selected. Following the Lagrange interpolations and FFT we thus end up with 5022 feature vectors representing the 5022 5-second segments of the 230 songs in a 4352-dimensional feature space. These feature vectors were further used to train a 22×22 dimensional *SOM*. Due to space restrictions we cannot provide a representation of the resulting map, yet we will use some examples for more detailed discussion.

For most songs the individual segments are mapped onto a rather small number of neighboring units. For example, we find most segments from classical titles mapped onto the lower right corner of the segment *SOM*. Some titles, such as *"Ironic"* by *Alanis Morissette* contain both rather soft and very dynamic passages and thus have their segments spread across several clusters co-located with segments from other songs of similar characteristics. However, the characteristics of some songs are too fuzzy to allow precise mapping of their segments and are thus spread across larger areas on the map.

In order to obtain a more compact representation of the musical archive, we create new feature vectors for each song based on the location of its segments. This results in a 22×22, i.e. 484-dimensional feature vector for each of the 230 songs. These vectors were used to train the 10×10 *SOM* presented in Figure 2.

Each song is now mapped onto one single position according to its musical characteristics. For example, we find a rather large cluster of classical music in the lower left corner of this map, including, amongst others, *Mozart's "Kleine Nachtmusik", Bach's "Air"* as well as the Andante of his *"Brandenburg Concerto No. 2"* on unit (0/8), next to the *"Moonlight Sonata"* by *Beethoven*. It is important to note, that the *SOM* does not organize the songs according to their melody, but rather according to their musical genre, i.e. their sound characteristics. We thus find mapped onto the same unit both *Tchaikovsky's "Schwanensee"* as well as *Bette Midler's "The Rose"*, a very soft love song with mostly Piano and Violin passages. Another example for this co-location of Pop and classic titles is *Madonna's "Frozen"*, located on the same unit as *Bach's "Fuge in D-Moll"* and the Overture of *Rossini's "Willhelm Tell"*.

To pick just one further example, we find *Cher's "Believe", Robbie Williams' "Rock DJ", The Pet Shop Boys' "Go West"* mapped together on unit (4/0) next to *Lou Bega's "Mambo No. 5"* and *Tom Jones' "Sexbomb"* on units (3/0) and (5/0), respectively.

4 Conclusions

We presented an approach to automatically organize music by content, i.e. based on its genre and sound characteristics. The *Self-Organizing Map*, a prominent unsupervised neural network, is used to cluster feature vectors representing the musical sound based on frequency spectra. In a first step, music segments are organized to obtain a fine-grained representation of segment-wise similarities, based upon which a clustering of the complete songs can be obtained. With this approach

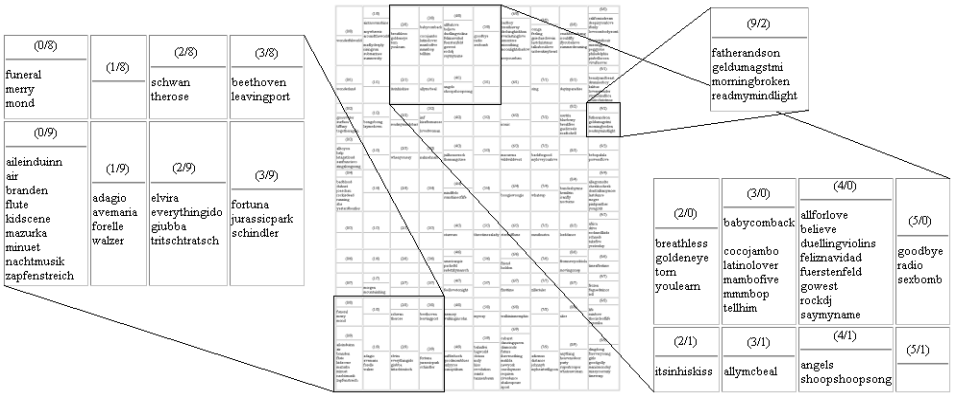

Figure 2: SOM representing 230 pieces of music

similar pieces of music are found in neighboring regions of the map. While the presented approach provides a good organization of music on the two-dimensional map, further improvements may be gained by capturing additional features during the vector creation process. These features may include beat information as well as representations capturing the dynamics of the various frequency bands. Furthermore, weighting functions may be used to assign higher importance to specific frequency bands.

References

[1] D. Bainbridge, C. Nevill-Manning, H. Witten, and R. McNab. Towards a digital library of popular music. In E. Fox and N. Rowe, editors, *Proc of the ACM Conf on Digital Libraries (ACMDL'99)*, pp 161–169, Berkeley, CA, 1999. ACM.

[2] M. Dittenbach, D. Merkl, and A. Rauber. The growing hierarchical self-organizing map. In *Intl Joint Conf on Neural Networks (IJCNN00)*, Como, Italy, 2000. IEEE.

[3] S. Dixon and E. Cambouropoulos. Beat tracking with musical knowledge. In *Proc of the Europ Conf on Artificial Intelligence*, Amsterdam, Netherlands, 2000.

[4] A. Ghias, J. Logan, D. Chamberlin, and S. B.C. Query by humming: Musical information retrieval in an audio database. In *Proc of the third ACM International Conf on Multimedia*, pp 231–236, San Francisco, CA, November 1995. ACM.

[5] T. Kohonen. *Self-organizing maps*. Springer-Verlag, Berlin, 1995.

[6] M. Melucci and N. Orio. Musical information retrieval using melodic surface. In*Proc of the ACM Conf on Digital Libraries (DL'99)*, Berkeley, CA, 1999. ACM.

[7] F. Nack and A. Lindsay. Everything you wanted to know about MPEG7 – part 1. *IEEE MultiMedia*, pp 65–77, July – September 1999.

[8] A. Rauber. SOMLib: A distributed digital library system based on self-organizing maps. In *Proc 10. Italian Workshop on Neural Nets (WIRN98)*, Vietri, Italy, 1998.