From Sound to 'Sense' via Feature Extraction and Machine Learning: Deriving High-Level Descriptors for Characterising Music

Gerhard Widmer[1,2], Simon Dixon[1], Peter Knees[2], Elias Pampalk[1], Tim Pohle[1]

[1] Austrian Research Institute for Artificial Intelligence, Vienna, Austria

[2] Department of Computational Perception, Johannes Kepler University Linz, Austria

1 Introduction

Research in intelligent music processing is experiencing an enormous boost these days due to the emergence of the new application and research field of *Music Information Retrieval (MIR)*. The rapid growth of digital music collections and the concomitant shift of the music market towards digital music distribution urgently call for intelligent computational support in the automated handling of large amounts of digital music. Ideas for a large variety of content-based music services are currently being developed in music industry and in the research community. They range from content-based music search engines to automatic music recommendation services, from intuitive interfaces on portable music players to methods for the automatic structuring and visualisation of large digital music collections, and from personalised radio stations to tools that permit the listener to actively modify and 'play with' the music as it is being played.

What all of these content-based services have in common is that they require the computer to be able to 'make sense of' and 'understand' the actual content of the music, in the sense of being able to recognise and extract musically, perceptually and contextually meaningful ('semantic') patterns from recordings, and to associate descriptors with the music that make sense to human listeners.

There is a large variety of musical descriptors that are potentially of interest. They range from low-level features of the sound, such as its bass content or its harmonic richness, to high-level concepts such as "hip hop" or "sad music". Also, semantic descriptors may come in the form of atomic, discrete labels like "rhythmic" or "waltz", or they may be complex, structured entities such as harmony and rhythmic structure. As it is impossible to cover all of these in one coherent chapter, we will have to limit ourselves to a particular class of semantic desciptors.

This chapter, then, focuses on methods for automatically extracting high-level atomic descriptors for the characterisation of music. It will be shown how high-level terms can be inferred via a combination of bottom-up audio descriptor extraction and the application of *machine learning* algorithms. Also, it will be shown that meaningful descriptors can be extracted not just from an analysis of the music (audio) itself, but also from extra-musical sources, such as the internet (via 'web mining').

Systems that learn to assign labels must be evaluated in systematic, controlled experiments. The most obvious and direct way is via *classification experiments*, where the labels to be assigned are interpreted as distinct classes. In particular, *genre classification*, i.e., the automatic assignment of an appropriate style label to a piece of music, has become a popular benchmark task in the MIR community (for many reasons, not the least of them being the fact that genre labels are generally much easier to obtain than other, more intuitive or personal descriptors). Accordingly, the current chapter will very much focus on genre classification as the kind of benchmark problem that measures the efficacy of machine learning (and the underlying descriptors) in assigning meaningful terms to music. However, in principle, one can try to predict any other high-level labels from low-level features, as long as there is a sufficient number of training examples with given labels. Some experiments regarding non-genre concepts will be briefly described in section 3.4, and in section 4.2 we will show how textual characterisations of music artists can be automatically derived from the Web.

The chapter is structured as follows. Section 2 deals with the extraction of music descriptors (both very basic ones like timbre and more abstract ones like melody or rhythm) from recordings via audio analysis. It focuses in particular on features that have been used in recent genre classification research. Section 3 shows how the gap between what can be extracted bottom-up and more abstract, human-centered concepts can be partly closed with the help of inductive machine learning. New approaches to infer additional high-level knowledge about music from extra-musical sources (the Internet) are presented in section 4. Section 5, finally, discusses current research and application perspectives and identifies important questions that will have to be addressed in the future.

2 Bottom-up Extraction of Descriptors from Audio

Extracting descriptors from audio recordings to characterise aspects of the audio content is not a new area of research. Much effort has been spent on feature extraction in areas like speech processing or audio signal analysis. It is impossible to give a comprehensive overview of all the audio descriptors developed over the past decades. Instead, this chapter will focus solely on descriptors that are useful for, or have been evaluated in, music classification tasks, in the context of newer work in Music Information Retrieval. The real focus of this chapter is on extracting or predicting higher-level descriptors via machine learning. Besides, a more in-depth presentation of audio and music descriptors is offered in another chapter of this book [REF. TO UPF CHAPTER], so the following sections only briefly recapitulate those audio features that have played a major role in recent music classification work.

Connected to the concept of classification is the notion of music or generally sound *similarity*. Obviously, operational similarity metrics can be used directly for audio and music classification (e.g., via nearest-neighbour algorithms), but also for a wide variety of other tasks. In fact, some of the music description schemes presented in the following do not produce features or descriptors at all, but directly compute similarities; they will also be mentioned, where appropriate.

2.1 Simple Audio Descriptors for Music Classification

This section describes some common simple approaches to describe properties of audio (music) signals. For all algorithms discussed here, the continuous stream of audio information is cut into small, possibly overlapping fragments of equal length, called *frames*. The typical length

of a frame is about 20 ms. Usually, for each frame one scalar value per descriptor is calculated, which can be done either on the time-domain or the frequency-domain representation of the signal. To obtain a (scalar) descriptor that pertains to an entire audio track, the values of all frames can be combined by, for example, applying simple statistics such as mean and standard deviation of all individual values.

2.1.1 Time-Domain Descriptors

On the time-domain representation of the audio signal, several descriptors can be calculated. An algorithm that mainly describes the power envelope of the audio signal is *Root Mean Square (RMS)*: The individual values appearing in each frame are squared, and the root of the mean of these values is calculated. These values might be combined as described above, or by calculating which fraction of all RMS values is below (e.g.) the average RMS value of a piece (*Low Energy Rate*). Comparable to the RMS values are the *Amplitude Envelope* values, which are the maximum absolute values of each frame. The amplitude envelope and RMS descriptors are commonly used as a first step in algorithms that detect rhythmic structure.

The time-domain representation might also be used to construct measures that model the concept of *Loudness* (i.e. the perceived "volume"). For example, a simple and effective way is to take the 0.23th power of the RMS values.

Another possibility is to approximately measure the perceived *brightness* with the *Zero Crossing Rate*. This descriptor simply counts how often the signal passes zero-level.

Also, the time-domain representation can be used to extract periodicity information from it. Common methods are autocorrelation and comb filterbanks. Autocorrelation gives for each given time lag the amount of self-similarity of the time domain samples by multiplying the signal with a time-lagged version of itself. In the comb filterbank approach, for each periodicity of interest, there is a comb filter with the appropriate resonance frequency.

2.1.2 Frequency-Domain Descriptors

A number of simple measures are commonly applied to describe properties of the frequency distribution of a frame:

- The *Band Energy Ratio* is the relation between the energy in the low frequency bands and the energy of the high frequency bands. This descriptor is vulnerable to producing unexpectedly high values when the energy in the low energy bands is close to zero.
- The *Spectral Centroid* is the center of gravity of the frequency distribution. Like the zero crossing rate, it can be regarded as a measure of perceived brightness or sharpness.
- The *Spectral Rolloff* frequency is the frequency below which a certain amount (e.g. 95%) of the frequency power distribution is concentrated.

These descriptors are calculated individually for each frame. The *Spectral Flux* is modeled to describe the temporal change of the spectrum. It is the Euclidean distance between the (normalised) frequency distributions of two consecutive frames, and can be regarded as a measure of the rate at which the spectrum changes locally.

The descriptors mentioned so far represent rather simple concepts. A more sophisticated approach are the *Mel Frequency Cepstral Coefficients (MFCCs)*, which model the shape of

the spectrum in a compressed form. They are calculated by representing the spectrum on the perceptually motivated Mel-Scale, and taking the logarithms of the amplitudes to simulate loudness perception. Afterwards, the discrete cosine transformation is applied, which results in a number of coefficients (MFCCs). Lower coefficients describe the coarse envelope of the frame's spectrum, and higher coefficients describe more detailed properties of the spectrum envelope. Usually, the higher-order MFCCs are discarded, and only the lower MFCCs are used to describe the music.

A popular way to compare two recorded pieces of music using MFCCs is to discard the temporal order of the frames, and to summarise them by clustering (e.g., [37, 2]). In the case of [2], for instance, the clustered MFCC representations of the frames are described by Gaussian Mixture Models (GMMs), which are the features for the piece of music. A way to compare GMMs is sampling: one GMM is used to produce random points with the distribution of this GMM, and the likelihood that the other GMM produces these points is checked.

It might seem that discarding the temporal order information altogether ignores highly important information. But recent research [15] has shown that MFCC-based description models using Hidden Markov Models (which explicitly model the temporal structure of the data) do not improve classification accuracy (as already noted in [5]), though they do seem to better capture details of the sound of musical recordings (at least in terms of statistical likelihoods). Whether this really makes a difference in actual applications remains still to be shown.

The interested reader is referred to Chapter XXX/UPF of this book for a much more comprehensive review of audio descriptors and music description schemes.

2.2 Extracting Higher-level Musical Patterns

The basic intuition behind research on classification by higher-level descriptors is that many musical categories can be defined in terms of high-level *musical* concepts. To some extent it is possible to define musical genre, for example, in terms of the melody, rhythm, harmony and instrumentation which are typical of each genre. Thus genre classification can be reduced to a set of subproblems: recognising particular types of melodies, rhythms, harmonies and instruments. Each of these subproblems is interesting in itself, and has attracted considerable research interest, which we review here.

Early work on music signal analysis is reviewed by Roads [45]. The problems that received the most attention were pitch detection, rhythm recognition and spectral analysis, corresponding respectively to the most important features of music: melody, rhythm and timbre (harmony and instrumentation).

Pitch detection is the estimation of the fundamental frequency of a signal, usually assuming it to be monophonic. Common methods include: time domain algorithms such as counting of zero-crossings and autocorrelation; frequency domain methods such as Fourier analysis and the phase vocoder; and auditory models which combine time and frequency domain information based on an understanding of human auditory processing. Recent work extends these methods to find the predominant pitch (usually the melody note) in polyphonic mixtures [20, 18].

The problem of extracting *rhythmic content* from a musical performance, and in particular finding the rate and temporal location of musical beats, has attracted considerable interest. A review of this work is found in [21]. Initial attempts focussed on rhythmic parsing of musical scores, that is without the tempo and timing variations that characterise performed

music, but recent tempo and beat tracking systems work quite successfully on a wide range of performed music. The use of rhythm for classification of dance music was explored in [11, 10].

Spectral analysis examines the time-frequency content of a signal, which is essential for extracting information about *instruments* and *harmony*. Short time Fourier analysis is the most widely used technique, but many others are available for analysing specific types of signals, most of which are built upon the Fourier transform. MFCCs, already mentioned in section 2.1.2 above, model the spectral contour rather than examining spectral content in detail, and thus can be seen as implicitly capturing the instruments playing (rather than the notes that were played). Specific work on instrument identification can be found in [25].

Regarding *harmony*, extensive research has been performed on the extraction of multiple simulateous notes in the context of automatic transcription systems, which are reviewed by Klapuri [27]. Transcription typically involves the follow steps: producing a time-frequency representation of the signal, finding peaks in the frequency dimension, tracking these peaks over the time dimension to produce a set of partials, and combining the partials to produce a set of notes. The differences between systems are usually related to the assumptions made about the input signal (for example the number of simultaneous notes, types of instruments, fastest notes, or musical style), and the means of decision making (for example using heuristics, neural nets or probabilistic reasoning).

Despite considerable successes, the research described above makes it increasingly clear that precise, correct, and general solutions to problems like automatic rhythm identification or harmonic structure analysis are not to be expected in the near future — the problems are simply too hard and would require the computer to possess the kind of broad musical experience and 'knowledge' that human listeners seem to apply so effortlessly when listening to music. Recent work in the field of Music Information Retrieval has thus started to focus more on *approximate* solutions to problems like melody extraction [14] or chord transcription [55], or on more *specialised* problems, like the estimation of global tempo [1] or tonality [17], or the identification of drum patterns [54].

Each of these areas provides a limited high level musical description of an audio signal. Systems have yet to be defined which combine all of these aspects, but this is likely to be seen in the near future.

3 Closing the Gap: Prediction of High-level Descriptors via Machine Learning

While the bottom-up extraction of features and patterns from audio continues to be a very active research area, it is also clear that there are strict limits as to the kinds of music descriptions that can be directly extracted from the audio signal. When it comes to intuitive, human-centered characterisations such as 'peaceful' or 'aggressive music' or highly personal categorisations such as 'music I like to listen to while working', there is little hope of analytically defining audio features that unequivocally and universally define these concepts. Yet such concepts play a central role in the way people organise and interact with and 'use' their music.

That is where *automatic learning* comes in. The only way one can hope to build a machine that can associate such high-level concepts with music items is by having the machine learn the correct associations between low-level audio features and high-level concepts, from examples of music items that have been labeled with the appropriate concepts. In this section, we give

a very brief introduction to the basic concepts of *machine learning* and *pattern classification*, and review some typical results with machine learning algorithms in musical classification tasks. In particular, the automatic labeling of music pieces with *genres* has received a lot of interest lately, and section 3.3 focuses specifically on genre classification. Section 3.4 then reports on recent experiments with more subjective concepts, which clearly show that a lot of improvement is still needed. One possible avenue towards achieving this improvement will then be discussed in section 4.

3.1 Classification via Machine Learning

Inductive learning as the automatic construction of classifiers from pre-classified training examples has a long tradition in several sub-fields of computer science. The field of *statistical pattern classification* [13, 23] has developed a multitude of methods for deriving classifiers from examples, where a 'classifier', for the purposes of this chapter, can be regarded as a black box that takes as input a new object to be classified (described via a set of features) and outputs a prediction regarding the most likely class the object belongs to. Classifiers are automatically constructed via *learning algorithms* that take as input a set of example objects labeled with the correct class, and construct a classifier from these that is (more or less) consistent with the given training examples, but also makes predictions on new, unseen objects — that is, the classifier is a *generalisation* of the training examples.

In the context of this chapter, training examples would be music items (e.g., songs) characterised by a list of audio features and labeled with the appropriate high-level concept (e.g., "this is a piece I like to listen to while working"), and the task of the learning algorithm is to produce a classifier that can predict the appropriate high-level concept for new songs (again represented by their audio features).

Common training and classification algorithms in statistical pattern classification [13] include nearest neighbour classifiers (k-NN), Gaussian Mixture Models, neural networks (mostly multi-layer feed-forward perceptrons), and support vector machines [9].

The field of Machine Learning [40] is particularly concerned with algorithms that induce classifiers that are *interpretable*, i.e., that explicitly describe the criteria that are associated with or define a given class. Typical examples of machine learning algorithms that are also used in music classification are decision trees [44] and rule learning algorithms [16].

Learned classifiers must be evaluated empirically, in order to assess the kind of prediction accuracy that may be expected on new, unseen cases. This is essentially done by testing the classifier on new (labeled) examples which have not been used in any way in learning, and recording the rate of prediction errors made by the classifier. There is a multitude of procedures for doing this, and a lot of scientific literature on advantages and shortcomings of the various methods. The basic idea is to set aside a part of the available examples for testing (the 'test set'), then inducing the classifier from the remaining data (the 'training set'), and then testing the classifier on the test set. A systematic method most commonly used is known as *n-fold cross-validation*, where the available data set is randomly split into n subsets ('folds'), and the above procedure is carried out n times, each time using one of the n folds for testing, and the remaining $n-1$ folds for training. The error (or conversely, accuracy) rates reported in most learning papers are based on experiments of this type.

A central issue that deserves some discussion is the *training data* required for learning. Attractive as the machine learning approach may be, it does require (large) collections of representative labeled training examples, e.g., music recordings with the correct categorisation

attached. Manually labeling music examples is a very laborious and time-consuming process, especially when it involves listening to the pieces before deciding on the category. Additionally, there is the copyright issue. Ideally, the research community would like to be able to share common training corpora. If a researcher wants to test her own features in classification experiment, she needs access to the actual audio files.

There are some efforts currently being undertaken in the Music Information Retrieval community to compile large repositories of labeled music that can be made available to all interested researchers without copyright problems. Noteworthy examples of this are Masataka Goto's RWC Music Database (http://staff.aist.go.jp/m.goto/RWC-MDB), the IMIRSEL (International Music Information Retrieval System Evaluation Laboratory) project at the University of Illinois at Urbana-Champaign (http://www.music-ir.org/evaluation — see also [12]), and the new FreeSound Initiative (http://freesound.iua.upf.edu).

3.2 Learning Algorithms Commonly Used in Music Classification

In this section, we briefly review some of the most common learning algorithms that are used in music classification and learning tasks.

Decision trees [44] are probably the most popular class of classification models in machine learning, and they are widely used also in Music Information Retrieval. In [49], for instance, decision tree learning algorithms have been used to build a model of the distribution of frame values.

Because of its known merits, k-NN classification is widely used. Sometimes, the feature values – possibly after feature selection – of each piece are regarded as a vector, and the distance used for k-NN classifier is the euclidean distance between individual pieces (e.g. [8, 22]) or to representative reference vectors (e.g. [24, 26]).

Support Vector Machines (SVMs) are also applied to music classification: e.g. [53] use them for genre classification, and [35] train several SVMs to recognise mood labels, where each SVM decides if one specific label is present in the music.

Gaussian Mixture Models (GMMs) are useful for estimating the distribution of feature values. They can be used as a classifier by modeling each class as a GMM; an instance is then classified by calculating, for each class (GMM), the likelihood that the instance was produced by the respective GMM, and predicting the class with the maximum likelihood. In [36], mood detection in classical music is done based on this approach. GMM classifiers have also been used in [7, 47] for genre classification.

Neural Networks have also been applied to music classification: [8] use a multilayer perceptron to determine the class of a piece given its feature vector. [24] use a more elaborate approach by training a separate neural network for each class, and an additional one that combines the outputs of these networks.

3.3 Genre Classification: Typical Experimental Results

The experimental results found in the literature on genre classification are not easy to compare, as researchers use many different music collections to evaluate their methods. Also, the ways of annotating the collections vary: some researchers label the pieces according to their own judgment, while others use online databases for the assignment of genre labels. Additionally, different authors often tackle slightly different problems (such as categorical

vs. probabilistic classification), which makes a comparison of the results even more difficult. These facts should be kept in mind when assessing the examples given in this section.

Generally, when trying to separate the classes Pop and Classical, very high accuracies are reached, suggesting that this task is not too difficult. E.g., [8] achieve up to 90.3% classification accuracy, and [38] report even 100% on 200 pieces. In both cases, the baseline is one half. Although [53] report a classification accuracy of 93% for four genres, in general the classification accuracy decreases when the number of genres grows.

For classification into dance music genres, [22] obtain up to 78,9% accuracy (15.9% baseline) when classifying 698 pieces of music into eight classes. This classification is based on a number of rhythmic descriptors and a rule-based classifier whose rules were designed manually. For a wider range of musical contents, divided into eleven genres, [48] report a classification accuracy of 67.6%, also based on rhythm features.

At the ISMIR 2004 conference, a comparison of different audio description algorithms was conducted in the form of a contest[1]. For the section of genre classification, the winning algorithm achieved a classification accuracy of 84.07% correct answers. The test collection consisted of 729 pieces, divided into six classes, with a baseline of 43.9%.

3.4 Trying to Predict Labels Other Than Genre

Genre or style is a descriptor that is useful for many applications, especially in commercial settings. Even though the concept of 'genre' is not well defined (see, e.g., [4]), it is still much more 'objective' than the kinds of personal characterisations human listeners attach to their music. But it is precisely these personal, subjective categorisations ("happy music", "aggressive music", "music I like whan I am sad", "music that one can dance to") that, if learnable by computers, would open new possibilities for intelligent and rewarding musical interactions between humans and machines.

A small preliminary experiment on the learnability of subjective, non-genre categorisations is reported in this section. As will be seen, the results are rather poor, and a lot of improvement is still needed. Web-based learning about music is a promising alternative that might help overcome the current limitations; that is the topic of the next section (Section 4).

The experiment presented here aimed to investigate the learnability of the categorisations *mood* (happy / neutral / sad), *perceived tempo* (very slow / slow / medium / fast / very fast / varying), *complexity* (low / medium / high), *emotion* (soft / neutral / aggressive), *focus* (vocal / both / instruments), and *genre* (blues / classical / electronica / folk / jazz / new age / noise / rock / world). To this end, each piece in a music collection of 729 pieces was labeled with the according value.

This data basis was used to examine the discriminative power of several descriptor sets in combination with a number of machine learning algorithms. The descriptor sets consisted mainly of descriptors that are widely used for music classification tasks (see section 2.1 above). Three different descriptor sets were tested: The set that was also used in [47], a set made from some Mpeg7 Low Level Descriptors, and a set that contained all descriptors of the above sets, together with some additional ones for rhythm and melody description.

To train the machine learning algorithms, mean and variance of the descriptors' values for a 30-second excerpt of the piece of music were taken as attributes. Table 1 shows the highest classification accuracies that were achieved with different learning algorithms; accuracy was

[1] http://ismir2004.ismir.net/ISMIR_Contest.html

estimated via stratified tenfold cross validation. The evaluated learning algorithms were J48 (a decision tree learner, available — like all the other learning algorithms mentioned here — in the machine learning toolkit WEKA[2]), SMO (a support vector machine), Naive Bayes, Naive Bayes with Kernel estimation, Boosting, Boosting with J48, Regression with MP5, Linear Regression, and k-NN with k = 1, 3, 5, 10. The table also lists the results obtained when applying the algorithm from [5] with to the same categorisations. For this algorithm, the best values obtained for k-NN classification with k = 1, 3, 5, 10 are shown. The other learning algorithms were not applicable to its feature data. Also, the baseline is given (i.e. the classification accuracy achieved when always guessing the most frequent class).

	mood	perceived tempo	complexity	emotion	focus	genre
Baseline	50.00 %	42.53 %	75.66 %	44.46 %	68.92 %	60.48 %
Set from [47]	50.00 %	42.53 %	76.63 %	45.06 %	71.08 %	65.66 %
Some Mpeg7 LLDs	50.00 %	43.13 %	76.14 %	46.75 %	70.00 %	64.94 %
"Large" Set	51.08 %	44.70 %	76.87 %	47.47 %	71.20 %	69.52 %
Best from [5]	50.24 %	48.67 %	78.55 %	57.95 %	75.18 %	70.84 %

Table 1: Best classification accuracies for the different categorisations in the small preliminary experiment.

These results show that with the examined techniques, in some cases it is even not possible to get classification accuracies higher than the baseline. For all categorisations except *mood*, the algorithm from [5] performed better than the other approaches. There is a number of ways in which this experiment could be improved, e.g., by the application of feature selection algorithms or the development of dedicated descriptors for each different task. Still, the results point to some fundamental limitations of the feature-based learning approach; concepts like the emotional quality of a piece of music seem to elude a purely audio-based approach.

4 A New Direction: Inferring High-level Descriptors from Extra-Musical Information

Listening to and 'making sense of' music is much more than decoding and parsing an incoming stream of sound waves into higher-level objects such as onsets, notes, melodies, harmonies, etc. Music is embedded in a rich web of cultural, historical, cultural, and social (and marketing) contexts that influence how music is heard, interpreted, and categorised. That is, many qualities or categorisations attributed to a piece of music by listeners cannot solely be explained by the content of the audio signal itself.

Also, recent research on genre classification is showing clearly that purely audio-based approaches to music classification may be hitting a kind of 'glass ceiling' [5]: there seem to be strict limits to the level of classification accuracy that can be obtained with purely audio-based features, no matter how sophisticated the audio descriptors. From a pragmatic point of view, then, it is clear that, if at all, high-quality automatic music annotation and classification can only be achieved by also taking into account and exploiting information sources that are external to the music itself.

[2]Software freely available from http://www.cs.waikato.ac.nz/ml/

The Internet is a rich, albeit unstructured, source of potential information, where millions of music lovers and experts discuss, describe, and exchange music. Possible information sources include personal web pages, music and concert reviews published on the Web, newspaper articles, discussion forums, chat rooms, playlists exchanged through peer-to-peer networks, and many more. A common term for denoting all the musically relevant information that is potentially "out there" is 'community metadata' [51]. Recent approaches to high-level music characterisation try to automatically extract relevant descriptors from the Internet — mostly from general, unstructured web pages —, via the use of information retrieval, text mining, and information extraction techniques (e.g., [6, 50, 51, 52]). In a sense, this is like learning about music without ever listening to it, by analysing the way people talk about and describe music, rather than what the music actually sounds like.

In the following, two research projects are briefly presented that show in a prototypical way how the Internet can be exploited as a source of information about – in this case – music artists. Section 4.1 shows how artists can be probabilistically related to genres via web mining, and section 4.2 presents an approach to the hierarchical clustering of music artists, and the automatic labeling of the individual clusters with descriptive terms gleaned from the Web.

4.1 Assigning Artists to Genres via Web Mining

In this section we will explain how to extract features (words) related to artists from web pages and how to use these features to construct a probabilistic genre classifier. This permits the computer to classify new artists present on the web using the Internet community's 'collective knowledge'. To learn the concept of a genre the method requires a set of typical artists for each genre in advance. Based on these artists and a set of web pages that talk about these artists, a characteristic profile is created for each genre. Using this profile (i.e. a weighted list of typical keywords) any artist can be classified by simple evaluation of word occurrences on related web pages. The following is a simplified account of the basic method; the details can be found in [29].

To obtain useful data for genre profile generation, Internet search engines like *Google* are queried with artist names, along with some constraints (e.g., *+music +review*) that should filter out non-musical pages, and the top ranked pages are retrieved. (Without these constraints, a search for groups such as *Kiss* would result in many unrelated pages). The retrieved pages tend to be common web pages such as fan pages, reviews from online music magazines, or music retailers. The first N available top-ranked webpages for each query are retrieved, all HTML markup tags are removed, so that only the plain text content is left, and common English stop word lists are used to remove frequent terms (e.g. a, and, or, the).

The *features* by which artists are characterised are the individual words that occur in any of the pages. In order to identify those words that may indicate what genre an artist belongs to, the next important step is feature weighting. A common method for this comes from the field of *Information Retrieval* and is known as term frequency $\times$ inverse document frequency ($tf \times idf$) [46]. For each artist a and each term t appearing in the retrieved pages, we count the number of occurrences tf_{ta} (term frequency) of term t in documents related to a, and df_t, the number of pages the term occurred in (document frequency). These are combined by multiplying the term frequency with the inverse document frequency. Basically, the intention of the $tf \times idf$ function is to assign a high score to terms that occur frequently, but also to reduce the score if these terms occur on many different pages and thus do not contain useful

information.

In the approach described in [29], an additional step is performed to find those terms that are most discriminative for each genre: a χ^2 test is used to select those terms that are least independent of (i.e., are likely to be predictive of) the classes. Selecting the top N terms for each category and scaling all χ^2 values per category such that the score for the top ranked term equals 1.0, gives a list of terms that seem to be typical of a given genre. An example of such a list for the genre heavy metal/hard rock is shown in Table 2. Note that neither of the constraint words (music and review) are included (they occur in all the pages, but they do not help in discriminating the genres).

The top 4 words are all (part of) artist names which were queried. However, many artists which are not part of the queries are also in the list, such as Phil Anselmo (Pantera), Hetfield, Hammett, Trujillo (Metallica), and Ozzy Osbourne. Furthermore, related groups such as Slayer, Megadeth, Iron Maiden, and Judas Priest are found as well as album names (Hysteria, Pyromania, ...) and song names (Paranoid, Unforgiven, Snowblind, St. Anger, ...) and other descriptive words such as evil, loud, hard, aggression, and heavy metal.

1.00 *sabbath	0.26 heavy	0.17 riff	0.12 butler
0.97 *pantera	0.26 ulrich	0.17 leaf	0.12 blackened
0.89 *metallica	0.26 vulgar	0.17 superjoint	0.12 bringin
0.72 *leppard	0.25 megadeth	0.17 maiden	0.12 purple
0.58 metal	0.25 pigs	0.17 armageddon	0.12 foolin
0.56 hetfield	0.24 halford	0.17 gillan	0.12 headless
0.55 hysteria	0.24 dio	0.17 ozzfest	0.12 intensity
0.53 ozzy	0.23 reinventing	0.17 leps	0.12 mob
0.52 iommi	0.23 lange	0.16 slayer	0.12 excitable
0.42 puppets	0.23 newsted	0.15 purify	0.12 ward
0.40 dimebag	0.21 leppards	0.15 judas	0.11 zeppelin
0.40 anselmo	0.21 adrenalize	0.15 hell	0.11 sandman
0.40 pyromania	0.21 mutt	0.15 fairies	0.11 demolition
0.40 paranoid	0.20 kirk	0.15 bands	0.11 sanitarium
0.39 osbourne	0.20 riffs	0.15 iron	0.11 *black
0.37 *def	0.20 s&m	0.14 band	0.11 appice
0.34 euphoria	0.20 trendkill	0.14 reload	0.11 jovi
0.32 geezer	0.20 snowblind	0.14 bassist	0.11 anger
0.29 vinnie	0.19 cowboys	0.14 slang	0.11 rocked
0.28 collen	0.18 darrell	0.13 wizard	0.10 drummer
0.28 hammett	0.18 screams	0.13 vivian	0.10 bass
0.27 bloody	0.18 bites	0.13 elektra	0.09 rocket
0.27 thrash	0.18 unforgiven	0.13 shreds	0.09 evil
0.27 phil	0.18 lars	0.13 aggression	0.09 loud
0.26 lep	0.17 trujillo	0.13 scar	0.09 hard

Table 2: The top 100 terms with highest χ^2_{tc} values for genre "heavy metal/hard rock" defined by 4 artists (Black Sabbath, Pantera, Metallica, Def Leppard). Words marked with * are part of the search queries. The values are normalised so that the highest score equals 1.0.

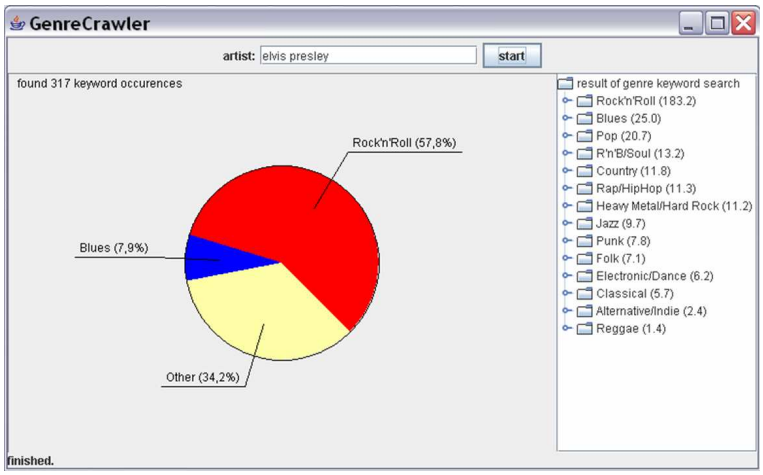

Figure 1: The GenreCrawler (cf. [29]) trying to classify Elvis Presley.

To classify previously unseen artists we simply query *Google* with the artist name, count the occurrences of the characteristic genre terms on the retrieved web pages, and multiply these numbers with their respective scores for each genre. The scores in each genre are summed up, and the probability of membership of an artist to a genre is then computed as the fraction of the achieved score of each genre over the sum of scores over all genres.

In [29], this procedure was tested using a genre taxonomy of 14 genres, and it was shown that correct genre recognition rates of 80% and better are achievable with this purely web-based approach, which compares very favourably with audio-based classification (see section 3.3 above).

On top of this classification system, an interactive demo applet (the "GenreCrawler") was implemented that permits the user to experiment with the system by typing in arbitrary new artists. In fact, the words to be typed in need not be artist names at all — they could be anything. The learned classifier can relate arbitrary words to genres, if that makes sense at all. For example, a query for "Pathétique" results in an unambiguous answer: *Classical Music*. A screenshot of the *GenreCrawler* at work can be seen in Figure 1.

4.2 Learning Textual Characterisations

It is easy to convert the linguistic features (words) identified with the above method into a *similarity measure*, again using standard methods from information retrieval. Similarity measures have a wide range of applications, and one is presented in this section: learning to group music artists into meaningful categories, and describing these categories with characteristic words. Again, this is exploiting the Internet as an information source and could not be achieved on an audio basis alone.

More precisely, the goal is to find words to describe what a group of artists has in common, or what distinguishes it from other groups. Such information can be used for hierarchical user interfaces to explore music collections an the artist level [42]. A simple text-based interface is shown in Figure 2 below.

As a first step, artists must be clustered hierarchically, and then appropriate terms (words) must be selected to describe these clusters. The basis of clustering is a similarity measure, which in our case is based on the linguistic features (characteristic words) extracted from Web

pages by the GenreCrawler. There is a multitude of methods for hierarchical clustering. In the system described here [42], basically, a one-dimensional self organising map (SOM) [31] is used, with extensions for hierarhical structuring [39, 32]. Overlaps between the clusters are permitted, such that an artist may belong to more than one cluster. To obtain a multi-level hierarchical clustering, for each cluster found another one-dimensional SOM is trained (on all artists assigned to the cluster) until the cluster size falls below a certain limit.

The second step is the selection of characteristic terms to describe the individual clusters. The goal is to select those words that best summarise a group of artists. The assumption underlying this application is that the artists are mostly unknown to the user (otherwise we could just label the clusters with the artists' names).

There are a number of approaches to select characteristic words [42]. One of these was developed by Lagus and Kaski (LK) [33] for labeling large document collections organised by SOMs. LK only use the term frequency tf_{ta} for each term t and artist a. The heuristically motivated ranking formula (higher values are better) is,

$$f_{tc} = (tf_{tc} / \sum_{t'} tf_{t'c}) \cdot \frac{(tf_{tc} / \sum_{t'} tf_{t'c})}{\sum_{c'} (tf_{tc'} / \sum_{t'} tf_{t'c'})}, \tag{1}$$

where tf_{tc} is the average term frequency in cluster c. The left side of the product is the importance of t in c defined through the frequency of t relative to the frequency of other terms in c. The right side is the importance of t in c relative to the importance of t in all other clusters.

To illustrate, Figure 2 shows a simple HTML interface that permits a user to explore the cluster structure learned by the system. There are two main parts to it: the hierarchy of clusters visualised as a grid of boxed texts and, just to the right of it, a display of a list of artists mapped to the currently selected cluster. The clusters of the first level in the hierarchy are visualised using the five boxes in the first (top) row. After the user selects a cluster, a second row appears which displays the children of the selected cluster. The selected clusters are highlighted in a different color. The hierarchy is displayed in such a way that the user can always see every previously made decision on a higher level. The number of artists mapped to a cluster is visualised by a bar next to the cluster. Inside a text box, at most the top 10 terms are displayed. The value of the ranking function for each term is coded through the color in which the term is displayed. The best term is always black and as the values decrease the color fades out. In the screenshot, at the first level the second node was selected, on the second level the fifth node, and on the third level, the first node. More details about method and experimental results can be found in [42].

To summarise, the last two sections were meant to illustrate how the Internet can be used as a rich source of information about music. These are just simple first steps, and a lot of research on extracting richer music-related information from the Web can be expected.

A general problem with web-based approaches is that many new and not so well known artists or music pieces do not appear on web pages. That limits the approach to yesterday's mainstream western culture. Another issue is the dynamics of web contents (e.g. [34]). This has been studied in [29] and the study was continued in [28]. The experiments reported there indicate that, while the web may indeed be unstable, simple approaches like the ones described here may be highly robust to such fluctuations in web contents. Thus, the web mining approach may turn out to be an important pillar in research on music categorisation, if not music 'understanding'.

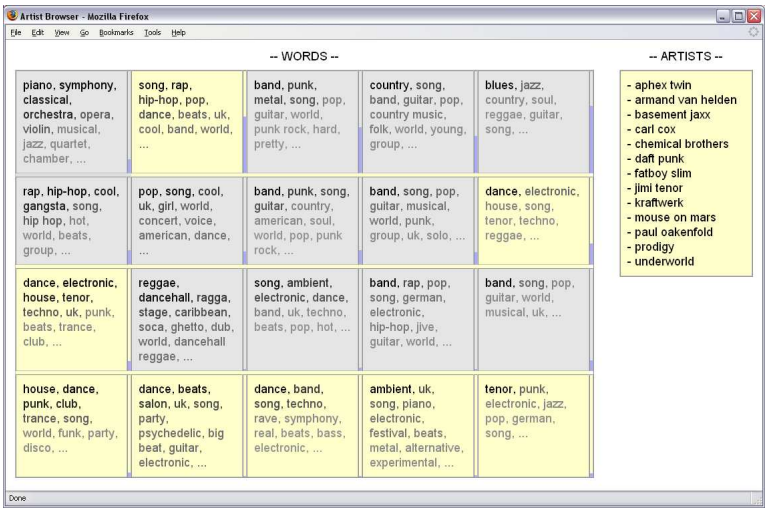

Figure 2: Screen shot of the HTML user interface to a system that automatically infers textual characterisations of artist clusters (cf. [42]).

5 Research and Application Perspectives

Building computers that can 'make sense' of music has long been a goal topic that inspired scientists, especially in the field of Artificial Intelligence (AI). For the past 20 or so years, research in AI and Music has been aiming at creating systems that could in some way mimic human music perception, or to put it in more technical terms, that could recognise musical structures like melodies, harmonic structure, rhythm, etc. at the same level of competence as human experts. While there has been some success in specialised problems such as beat tracking, most of the truly complex musical capabilities are still outside of the range of computers. For example, no machine is currently capable of correctly transcribing an audio recording of even modest complexity, or of understanding the high-level *form* of music (e.g., recognising whether a classical piece is in sonata form, identifying a motif and its variations in a Mozart sonata, or unambiguously segmenting a popular piece into verse and chorus and bridge).

The new application field of Music Information Retrieval has led to, or at least contributed to, a shift of expectations: from a practical point of view, the real goal is not so much for a computer to 'understand' music in a human-like way, but simply to have enough 'intelligence' to support intelligent musical services and applications. Perfect musical understanding may not be required here. For instance, genre classification need not reach 100% accuracy to be useful in music recommendation systems. Likewise, a system for quick music browsing (e.g., [19]) need not perform a perfect segmentation of the music — if it finds roughly those parts in a recording where some of the interesting things are going on, that may be perfectly sufficient. Also, relatively simple capabilities like classifying music recordings into broad categories (genres) or assigning other high-level 'semantic' labels to pieces can be immensely useful.

As has been indicated in this chapter, some of these capabilities are within reach, and indeed, some highly interesting real-world applications of this technology are currently emerging in the music market. From the research point of view, it is quite clear that there is still

ample room for improvement, even within the relatively narrow domain of learning to assign high-level descriptors and labels to music recordings, which was the topic of this chapter. For instance, recent work on musical web mining has shown the promise of using extra-musical information for music classification, but little research has so far been performed on *integrating* different information sources — low-level audio features, higher-level structures automatically extracted from audio, web-based features, and possibly lyrics (which can also be recovered automatically from the Internet [30]) — in non-trivial ways.

A concept of central importance to MIR is *music similarity measures*. These are useful not only for classification, but for a wide variety of practical application scenarios, e.g., the automatic structuring and visualisation of large digital music collections [43, 41], automatic playlist generation (e.g., [3]), automatic music recommendation, and many more. Current music similarity measures are usually based on lower-level descriptors which are somehow averaged over a whole piece, so that a Euclidean distance metric can be applied to them. More complex approaches like clustering and distribution modelling via mixtures give a slightly more detailed account of the contents of a piece, but still ignore the temporal aspect of music. While preliminary experiments with Hidden Markov Models [5, 15], which do model temporal dependencies, do not seem to lead to improvements when based on low-level timbral features (like MFCCs), there is no reason to assume that the integration of higher-level descriptors (like melody, harmony, etc.) and temporal modelling will not permit substantial improvement. A lot of research on these issues is to be expected in the near future, driven by the sheer practical potential of music similarity measures. To put it simply: computers equipped with good music similarity measures may not be able to *make sense* of music in any human-like way, but they will be able to do more and more *sensible things* with music.

Acknowledgments

This work is supported by the European Union in the context of the projects S2S^2 ("Sound to Sense, Sense to Sound", IST-2004-03773) and SIMAC ("Semantic Interaction with Music Audio Contents", FP6 507142). Further support for ÖFAI's research in the area of intelligent music processing is currently provided by the following institutions: the European Union (project COST 282 KnowlEST "Knowledge Exploration in Science and Technology"); the Austrian *Fonds zur Förderung der Wissenschaftlichen Forschung* (FWF; projects Y99-START "Artificial Intelligence Models of Musical Expression" and L112-N04 "Operational Models of Music Similarity for MIR"); and the Viennese *Wissenschafts-, Forschungs- und Technologiefonds* (WWTF; project CI010 "Interfaces to Music"). The Austrian Research Institute for Artificial Intelligence also acknowledges financial support by the Austrian Federal Ministries of Education, Science and Culture and of Transport, Innovation and Technology.

References

[1] M. Alonso, B. David, and G. Richard. Tempo and beat estimation of musical signals. In *5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 158–163, 2004.

[2] J.J. Aucouturier and F. Pachet. Music similarity measures: What's the use? In *Proceedings of the Third International Symposium on Music Information Retrieval (ISMIR 2002)*, pages 157–163, Paris, France, 2002.

[3] J.J. Aucouturier and F. Pachet. Scaling up music playlist generation. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME 2002)*, Lausanne, Switzerland, 2002.

[4] J.J Aucouturier and F. Pachet. Musical genre: A survey. *Journal of New Music Research*, 32(1):83–93, 2003.

[5] J.J. Aucouturier and F. Pachet. Improving timbre similarity: How high is the sky? *Journal of Negative Results in Speech and Audio Sciences*, 1(1), 2004.

[6] S. Baumann and O. Hummel. Using cultural metadata for artist recommendation. In *Proceedings of the International Conference on Web Delivery of Music (WedelMusic)*, Leeds, UK, 2003.

[7] J.-J. Burred and A. Lerch. A hierarchical approach to automatic musical genre classification. In *Proceedings of th 6th International Conference on Digital Audio Effects (DAFx-03), London, UK, September 8-11, 2003*, London, UK, September 8-11 2003.

[8] C. H. L. Costa, J. D. Valle Jr., and A. L. Koerich. Automatic classification of audio data. In *Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics - SMC*, Hague, Netherlands, October, 10-13 2004.

[9] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[10] S. Dixon, F. Gouyon, and G. Widmer. Towards characterisation of music via rhythmic patterns. In *5th International Conference on Music Information Retrieval*, pages 509–516, 2004.

[11] S. Dixon, E. Pampalk, and G. Widmer. Classification of dance music by periodicity patterns. In *4th International Conference on Music Information Retrieval*, pages 159–165, 2003.

[12] J.S. Downie, J. Futrelle, and D. Tcheng. The international music information retrieval systems evaluation laboratory: Governance, access and security. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04)*, Barcelona, Spain, 2004.

[13] R. Duda, P Hart, and D. Stork. *Pattern Classification (2nd Edition)*. John Wiley & Sons, New York, 2001.

[14] J. Eggink and G. Brown. Extracting melody lines from complex audio. In *5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 84–91, 2004.

[15] A. Flexer, E. Pampalk, and G. Widmer. Hidden markov models for spectral similarity of songs. In *Submitted*, 2005.

[16] J. Fürnkranz. Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54, 1999.

[17] E. Gómez and P. Herrera. Extimating the tonality of polyphonic audio files: Cognitive versus machine learning modelling strategies. In *5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 92–95, 2004.

[18] E. Gómez, A. Klapuri, and B. Meudic. Melody description and extraction in the context of music content processing. *Journal of New Music Research*, 32(1):23–41, 2003.

[19] M. Goto. Smartmusickiosk: Music listening station with chorus-search function. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST 2003)*, pages 31–40, 2003.

[20] M. Goto and S. Hayamizu. A real-time music scene description system: Detecting melody and bass lines in audio signals. In *Working Notes of the IJCAI-99 Workshop on Computational Auditory Scene Analysis*, pages 31–40. International Joint Conference on Artificial Intelligence, 1999.

[21] F. Gouyon and S. Dixon. A review of automatic rhythm description systems. *Computer Music Journal*, 29(1):34–54, 2005.

[22] F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer. Evaluating rhythmic descriptors for musical genre classification. In *Proceedings of the AES 25th International Conference*, London, UK, June 17-19 2004.

[23] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Verlag, New York, 2001.

[24] O. Hellmuth, E. Allamanche, J. Herre, T. Kastner, N. Lefebvre, and R. Wistorf. Music genre estimation from low level audio features. In *Proceedings of the AES 25th International Conference*, London, UK, June 17-19 2004.

[25] P. Herrera, G. Peeters, and S. Dubnov. Automatic classification of musical instrument sounds. *Journal of New Music Research*, 32(1):3–22, 2003.

[26] T. Kastner, J. Herre, E. Allamanche, O. Hellmuth, C. Ertel, and M. Schalek. Automatic optimization of a music similarity metric using similarity pairs. In *Proceedings of the AES 25th International Conference*, London, UK, June 17-19 2004.

[27] A. Klapuri. Automatic music transcription as we know it today. *Journal of New Music Research*, 33(3):269–282, 2004.

[28] P. Knees. Automatische Klassifikation von Musikkünstlern basierend auf Web-Daten (automatic classification of music artists based on web-data). Masters thesis, Vienna University of Technology, Vienna, 2004.

[29] P. Knees, E. Pampalk, and G. Widmer. Artist classification with web-based data. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04)*, Barcelona, Spain, 2004.

[30] P. Knees, M. Schedl, and G. Widmer. Multiple lyrics alignment: Automatic retrieval of song lyrics. In *Submitted*, 2005.

[31] T. Kohonen. *Self-Organizing Maps*. Springer Verlag, 2001.

[32] P. Koikkalainen and E.Oja. Self-organizing hierarchical feature maps. In *Proceedings of the International Joint Conference on Neural Networks*, San Diego, CA, 1990.

[33] K. Lagus and S. Kaski. Keyword selection method for characterizing text document maps, volume 1. In *Proceedings of ICANN99, Ninth International Conference on Artificial Neural Networks*, pages 371–376, London, 1999. IEEE.

[34] S. Lawrence and C. Lee Giles. Accessibility of information on the web. *Nature*, 400(6740):107–109, 1999.

[35] T. Li and M. Ogihara. Detecting emotion in music. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR'03)*, Baltimore, MD, USA, October 26-30 2003.

[36] D. Liu, L. Lu, and H.-J. Zhang. Automatic mood detection from acoustic music data. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR'03)*, Baltimore, MD, USA, October 26-30 2003.

[37] B. Logan and A. Salomon. A music similarity function based on signal analysis. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2001.

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

[39] R. Miikkulainen. *Script recognition with hierarchical feature maps*. Connection Science, 1990.

[40] T.M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[41] E. Pampalk, S. Dixon, and G. Widmer. Exploring music collections by browsing different views. *Computer Music Journal*, 28(2):49–62, 2004.

[42] E. Pampalk, A. Flexer, and G. Widmer. Hierarchical organization and description of music collections at the artist level. In *Proceedings of the 9th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2005)*, 2005.

[43] E. Pampalk, A. Rauber, and D. Merkl. Content-based organization and visualization of music archives. In *Proceedings of the 10th ACM International Conference on Multimedia*, pages 570–579, Juan les Pins, France, 2002.

[44] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[45] C. Roads. *The Computer Music Tutorial*. MIT Press, Cambridge MA, 1996.

[46] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

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

[48] C. Uhle and C. Dittmar. Drum pattern based genre classification of popular music. In *Proceedings of the AES 25th International Conference*, London, UK, June 17-19 2004.

[49] K. West and S. Cox. Features and classifiers for the automatic classification of musical audio signals. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR'04)*, Barcelona, Spain, October, 10-14 2004.

[50] B. Whitman and D. Ellis. Automatic record reviews. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, Barcelona, Spain, 2004.

[51] B. Whitman and S. Lawrence. Inferring descriptions and similarity for music from community metadata. In *Proceedings of the 2002 International Computer Music Conference (ICMC 2002)*, Göteborg, Sweden, 2002.

[52] B. Whitman and B. Smaragdis. Combining musical and cultural features for intelligent style detection. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR 2002)*, Paris, France, 2002.

[53] C. Xu, N. C. Maddage, X. Shao, and F. C. Tian. Musical genre classification using support vector machines. In *Proceedings of IEEE ICASSP03*, Hong Kong, China, April 6-10 2003.

[54] K. Yoshii, M. Goto, and H. Okuno. Automatic drum sound description for real-world music using template adaptation and matching methods. In *5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 184–191, 2004.

[55] T. Yoshioka, T. Kitahara, K. Komatani, T. Ogata, and H. Okuno. Automatic chord transcription with concurrent recognition of chord symbols and boundaries. In *5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 100–105, 2004.