Journal of New Music Research
2003, Vol. 32, No. 1, pp. 83–93

0929-8215/03/3201-083$16.00

Representing Musical Genre: A State of the Art

Jean-Julien Aucouturier and François Pachet

SONY Computer Science Laboratory, Paris, France

Abstract

Musical genre is probably the most popular music descriptor. In the context of large musical databases and Electronic Music Distribution, genre is therefore a crucial metadata for the description of music content. However, genre is intrinsically ill-defined and attempts at defining genre precisely have a strong tendency to end up in circular, ungrounded projections of fantasies. Is genre an intrinsic attribute of music titles, as, say, tempo? Or is genre a extrinsic description of the whole piece? In this article, we discuss the various approaches in representing musical genre, and propose to classify these approaches in three main categories: manual, prescriptive and emergent approaches. We discuss the pros and cons of each approach, and illustrate our study with results of the Cuidado IST project.

1. Introduction

The new context of Electronic Music Distribution and systematic exploitation of large musical databases creates a need to produce symbolic descriptions of music titles. Musical genre is probably the most obvious descriptor which comes to mind, and it is probably the most widely used form of music description. However, genre is intrinsically ill-defined and attempts at defining precisely genre have a strong tendency to end up in circular, ungrounded projections of fantasies. Genre is intrinsically related to classification: ascribing a genre to an item is indeed a useful way of describing what this items shares with other items – of the same genre – and also what makes this item different from items – of other genres. The genesis of genre is therefore to be found in our natural and irrepressible tendency to classify.

This is not good news. Genre suffers from an intrinsic ambiguity, deeply rooted in our dualist view of the world. First, genre may be used as an *intentional* concept. In this view, genre is an interpretation of a title, produced and possibly shared by a given community, much in the same way we ascribe and interpret meanings to words in our languages. Genre is here a linguistic category, useful for instance to talk about music titles: *Yesterday* by the Beatles is a "Brit-Pop" title, because it is by the Beatles, and we all share cultural knowledge about this group, the 60s, etc.

Alternatively, genre may be used as an *extensional* concept. In this view, genre are sets of music titles. Here, genre is closely related to analysis: genre is a dimension of a music title, much like tempo, timbre or the language of the lyrics. *Yesterday* by the Beatles is a mellow Pop Song, because it has a cheesy medium tempo melody, string backup and it is sung with a melancholic voice.

In idealistic, mathematical worlds, intentional and extensional definitions coincide. In the real world they do not, so no unique position can be taken regarding genre. The aim of this article is to review the various approaches to represent musical genre explicitly, and to discuss the pros and cons of each approach. More precisely, we propose to classify these approaches in three main categories. Manual efforts consist on representing human expert knowledge about music titles. Approaches aiming at extracting genre automatically are themselves divided into two different categories, depending on what is considered as objective and what is not. On the one hand, prescriptive approaches attempt to model existing genre classifications as they are found. These models are based on combinations of low-level features of the signal. On the other hand, emergent approaches aim at building genre taxonomies grounded in objective similarity measures.

The survey is structured as follows. In the first section of this review, we compare and discuss existing classifications of musical genre. In particular, we describe the process of manually classifying music titles in the framework of the European project Cuidado (Pachet, 2001). We then review the automatic approaches. The following section is devoted

Accepted: 9 May, 2002

Correspondence: Jean-Julien Aucouturier, SONY Computer Science Laboratory, 6, rue Amyot, 75005 Paris, France. Tel.: +33 144080513, E-mail: jj@csl.sony.fr

to *prescriptive* approaches, based on supervised classification techniques on a space of low-level timbral features, given arbitrary taxonomies. The final section is devoted to approaches in which the classification *emerges* given arbitrary similarity measures. We describe in particular a data-mining technique to extract similarity which appears to be well-suited to genre clustering.

2. Genre taxonomies and manual classification

2.1 Analysis of existing genre classifications

Musical Genre is widely used to classify and describe titles, both by the music industry and the consumers. There are therefore many different genre taxonomies available. The authors have done a thorough study of the existing genre taxonomies used in different official music resources, including:

- Record Company catalogues: Universal, Sony Music, EMI, BMG
- Record shops and megastores: Virgin Megastore, Tower Records, Fnac . . .
- Music charts: Billboard, Top 50, Cashbox . . .
- Musical web sites and online record shops: Amazon, All Music, SonicNet, Mzz, Listen, Netbeat . . .
- Specialized press and books
- Specialized web radios

The complete analysis can be found in Pachet and Cazaly (2000). Here, we summarize the main conclusions.

Taxonomies for albums or for titles?

Most taxonomies in use today are album-oriented: the music industry, so far, sells albums, and not individual titles. This constraint obviously has an impact on the structure of the taxonomy, since albums often contain titles of many different genres. Moreover, one cannot simply expand album-oriented taxonomies to describe music titles. Album-oriented taxonomies are utterly inappropriate for describing titles, except for large categories (Classical versus Rock for instance). They simply do not describe the same objects.

No consensus

Pachet and Cazaly (2000) compares 3 Internet genre taxonomies: allmusic.com (531 genres), amazon.com (719 genres) and mp3.com (430 genres). Results show that there is no consensus in the name used in these classifications: only 70 words are common to the three taxonomies. More importantly, there is no shared structure: among these common words, even largely used terms like "Rock" or "Pop" do not have common definitions, i.e., they do not denote the same set of songs. Finally, the location of these taxons (i.e., nodes) in the hierarchy differs from one taxonomy to the other. Semantic interoperability of "natural" genre taxonomies is clearly a dream.

Semantic inconsistency within taxonomies

Within given taxonomies, it has been shown that taxons do not bear constant, fixed semantics. For instance, a taxonomy such as the one used by Amazon contains taxons denoting period ("60s pop"), topics ("love song"), country ("Japanese music"), language ("French Variety"), dance types ("Waltz"), artist Type ("Crooner") . . . Classifications often oscillate between these different interpretations (e.g., Jazz/Blues/Delta/Pre-War).

This semantic confusion leads to many redundancies in the taxonomy, and it is obviously a poor description scheme for automatic systems. It is important to notice that this confusion, however, has apparently no impact on the efficiency of the taxonomy for human users. It is indeed easy to navigate in these taxonomies, and switching semantics at each taxonomic level is natural for most users.

What do labels mean?

A great deal of information is implicitly contained in taxon labels. For instance, the music style "World Italian" may contain all Italian artists as well as artists that sing in Italian, not necessarily sharing any stylistic similarity (from Opera singer Pavarotti to a metal band from Milan).

This study illustrates that music genre is an ill-defined notion, that is not founded on any *intrinsic* property of the music, but rather depends on cultural *extrinsic* habits (e.g., to a French music lover, singer Charles Aznavour could be classified as "Variety," but in record shops in the UK, it is filed under "World Music").

2.2 Manual genre classification

However arbitrary, genre classifications are deeply needed. Virtually all EMD projects devote a substantial part of their time in the design of genre taxonomies. For instance, Weare in Dannenberg et al. (2001) describes the manual effort in building a genre taxonomy for Microsoft's MSN Music Search Engine.

These efforts often take gigantic proportions. Weare states that the manual labeling of a "few hundred-thousand songs" for Microsoft MSN required musicologists to be brought as full-time employees and took about 30 man-years. The details of the taxonomy and the design methodology are, however, not available.

In the CUIDADO project, we have initially taken this route and built a rather detailed taxonomy of genre for music titles, described in Pachet and Cazaly (2000). In order to limit the aforementioned lexical and semantic inconsistencies, we chose to consider genre as an independent descriptor, compared to the other descriptors in our metadatabase. The other descriptors contained the most frequent criteria used in music classifications ("country," "instrumentation," "artist type," etc.). Combining genre with these other attributes allowed us to limit the explosion of musical genres: instead of defining

"French-Rock," "English-Rock, "Spanish-Rock," we would only define the genre "Rock" and use the "country" attribute.

Additionally our genre taxonomy included similarity relations, either based on:

- inheritance: "Rock/Alternatif/Folk" ↔ "Rock/Alternatif/Ska"
- string-matching: "Rock/Latino" ↔ "World/Latino"
- expert knowledge: explicit links across stylistic regions, e.g., "Rhythm&Blues/Tamla Motown" ↔ "Soul/Disco/Philadelphia" because they have the same orchestration (brass and strings).

However, we eventually decided to give up this effort for the following reasons:

- The bottom taxons were very difficult to describe objectively. Only the taxonomy designers would be able to distinguish between slightly different taxons. For instance, the taxons labeled Rock-California would differ from Rock-FM only by the fact that Rock-California titles would contain predominant "acoustic guitar" sounds. Although there are strong arguments to enforce this distinction, it is also clear that these arguments are not easily shareable.
- The taxonomy was very sensitive to music evolution. Music evolution has notable and well-known effects in genre taxonomies. New genres appear frequently (e.g., Trip-Hop, Acid-Jazz, Post-Rock), and these genres are often very difficult to insert in an existing taxonomy, notably because of multiple inheritance issues ("Jazz-Reggae"). Furthermore, music evolution induces phenomenons of genre compression (different genres are merged) and expansion (genres split into subgenres). Precise taxonomies of this kind are not only difficult to build, but also impossible to maintain.

Our experiment is by no means a proof of impossibility, but we have chosen to reduce our ambitions, and focused on developing a much simpler genre taxonomy, aiming at describing artists rather than titles. This choice can be seen as a trade-off between precision (The Beatles have mostly done music with a vocal part, but some of their songs are instrumental, for instance *Revolution 9*) and scalability (there are far fewer artists than songs).

This genre-of-artist attribute is stored in an 12000 artist database, together with other features such as Name ("The Beatles"), Interpretation ("Vocal," "Instrumental"), Language ("English"), Type ("band," "man"), . . .

In any case, manual input is clearly not sufficient to describe precisely millions of titles. Manual input is therefore mostly useful as a bootstrap to test research ideas, or as a comparison base to evaluate automatic algorithms.

In the next two sections, we identify and review two approaches to Automatic Musical Genre Classification. The first approach is *prescriptive*, as it tries to classify songs in an arbitrary taxonomy, given a priori. The second approach adopts a reversed point-of-view, in which the classification *emerges* from the songs.

3. The prescriptive approach to automatic musical genre classification

There have been numerous attempts at extracting genre information automatically from the audio signal, using signal processing techniques and machine learning schemes. We review here 8 recent contributions (Tzanetakis & Cook, 2000a; Tzanetakis et al., 2001; Talapur et al., 2000; Pye, 2000; Soltau, 1998; Lambrou & Sandler, 1998; Deshpande et al., 2001; Ermolinskiy et al., 2001) and compare the algorithms used. All of these works make the same assumption that a genre taxonomy is given and should be superimposed on the database of songs (as seen before, such a taxonomy is in fact arbitrary). They all proceed in two steps:

- Frame-based Feature extraction: the music signal is cut into frames, and a feature vector of low-level descriptors of timbre, rhythm, etc. is computed for each frame.
- Machine Learning/Classification: a classification algorithm is then applied on the set of feature vectors to label each frame with its most probable class: its "genre." The class models used in this phase are trained beforehand, in a supervised way.

3.1 Feature extraction

The features used in the first step of automatic, prescriptive genre classification systems can be classified in 3 sets: timbre related, rhythm related and pitch related.

Timbre related

Most of the feature used in genre classification systems describe the spectral distribution of the signal, i.e., a global "timbre." Here "global" means that it encompasses all the sources and instruments in the music, but does not mean that only one timbre value is computed for the whole song to be classified: timbre features are extracted from every frame.

- *FFT coefficients* (used in Tzanetakis and Cook, 2000a; Talapur et al., 2000; Deshpande et al., 2001):
  For each frame, the feature vector is simply the vector of the 128, 256, etc. FFT coefficients.
- *Cepstrum and Mel Cepstrum Coefficients (MFCC)* (used in Tzanetakis and Cook (2000a); Pye (2000); Soltau (1998); Deshpande et al. (2001)):
  The cepstrum is the inverse Fourier transform of the log-spectrum log(*S*).

$$Cn = \frac{1}{2\pi}\int_{\omega=-\pi}^{\omega=\pi} \log S(\omega)\exp(j\omega n)d\omega$$

  We call mel-cepstrum the cepstrum computed after a non-linear frequency warping onto a perceptual frequency

scale, the Mel-frequency scale (Rabiner & Juang, 1993). The c_n are called Mel frequency cepstrum coefficients (MFCC). Cepstrum coefficients provide a low-dimensional, smoothed version of the log spectrum, and thus are a good and compact representation of the spectral shape. They are widely used as feature for speech recognition, and have also proved useful in musical instrument recognition (Eronen & Klapuri, 2000).

– *Linear Prediction (LP)* (used in Tzanetakis and Cook (2000a)): This echoes a classic model for sound production, in which the observed data results from a source signal passing through a linear filter:

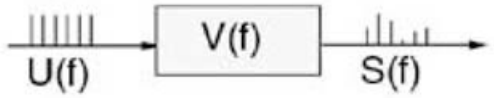

Fig. 1. The source-filter model of sound production.

Although it has been designed to model speech production, it is also partially valid for musical instruments: in this case, the source signal is a periodic impulse that includes the pitch information, and the filter $V(z)$ embodies the effect of the resonating body of the instrument; namely, its timbre.

With Linear Prediction we estimate the coefficients of the filter $V(z)$, assuming it is all-pole of order p.

$$V(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_p z^{-p}}$$

where a_i are the filter coefficients. They control the position of the poles of the transfer function $V(z)$, i.e., the position of the peaks on the power spectral density of the signal. Therefore, like MFCC, LP is a way to encode the spectral envelope of the signal.

– *MPEG filterbank components* (Tzanetakis & Cook, 2000a; Pye, 2000):

To compute the previous features, one usually uses raw audio, such as .wav files. However, the huge majority of music files available for analysis are compressed using the MPEG audio compression standard, thus they have to be first decompressed into wav files. One interesting possibility for speeding computation is to calculate the features directly from the mpg data. This idea has been proposed by Tzanetakis and Cook (2000b), and notably implemented by Pye (2000).

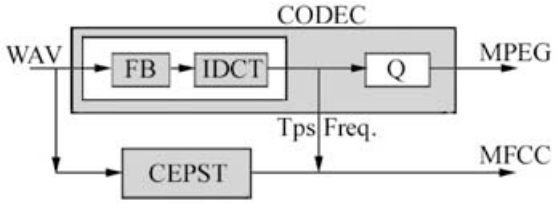

Fig. 2. Comparison of the MPEG compression and the MFCC calculation.

Figure 2 compares the processing involved in the MPEG compression (upper path) and the MFCC computation (bottom path). On the one hand, in MPEG-1 layer 3 compression (known as MP3), the signal is first converted to spectral components via an analysis filterbank (each filter reproduces one of the cochlea's critical bands), and further subdivided in frequency content by applying a 6-point or 18-point modified DCT block transform. Then, each spectral component is quantized and coded with the goal of keeping the quantization noise below the masking threshold, estimated via a psycho-acoustic model. More details can be found in (ISO/IEC). On the other hand, MFCC computation requires the same first two steps: Filtering on a Mel-Frequency scale, and taking a DCT to decorrelate the coefficients. Thus, it seems advantageous not to decompress MPEG into .wav before computing the MFCC, because this amounts to partly re-doing the same process backwards. Instead, it is easy to bypass the frequency analysis by just inverting the bit-allocation algorithm.

– *Spectral Centroid* (Tzanetakis et al., 2001; Lambrou & Sandler, 1998):

The Spectral Centroid is the barycentre point of the spectral distribution within a frame.

$$SC = \frac{\sum_k kS(k)}{\sum_k S(k)}$$

where S is the magnitude spectrum of a frame. This feature gives an indication of the spectral shape, and is classically used in monophonic instrument recognition (see for instance Martin, 1998).

– *Spectral Flux* (Tzanetakis et al., 2001):

This feature measures frame-to-frame spectral difference, i.e., the change in the spectral shape. This is also a classic feature used in monophonic instrument recognition.

$$SF = \| S_{[e]} - S_{[e-1]} \|$$

where l is the frame number and S the complete magnitude spectrum of a frame.

– *Zero Crossing Rate* (Tzanetakis et al., 2001; Lambrou & Sandler, 1998):

ZCR is the number of time-domain zero crossings within a frame. It is a measure of the noise behavior of the signal, and a correlate of the pitch.

– *Spectral Roll-Off* (Tzanetakis et al., 2001):

The Roll-Off point is the frequency below which some percentage (e.g., 85%) of the power of the spectrum resides. This decreases with percussive sounds and attach transients.

$$SR = L \text{ so that } \sum_{k<L} S(k) = 0.85 \times \sum_k S(k)$$

– *Low order statistics* (Tzanetakis & Cook, 2000a; Lambrou & Sandler, 1998; Talapur et al., 2000; Tzanetakis et al., 2001):

Many authors also compute the low order statistics of the previous features, over larger analysis windows: mean (1st order), variance (2nd), skewness (3rd) and kurtosis (4th). Note that mathematically, one can rebuild the spectral distribution from the infinite series of its moments, hence the two representations are theoretically equivalent.

– *Delta-coefficients* (Tzanetakis & Cook, 2000a; Talapur et al., 2000):
In an attempt to take account of the dynamics of the data, it is common practice to append delta coefficients to the feature set, i.e., differenced coefficients that measure the change of coefficients between consecutive frames.

Time and rhythm related

Some authors suggest that a genre classifier should not just rely on "global timbre" descriptors, but also take rhythm into account. Tzanetakis et al. (2001) uses a "beat histogram" built from the autocorrelation function of the signal: by looking at the weight of the different periodicities in the signal (and the ratios between these weights), one has an idea of the "strongness" and complexity of the beat in the music. This is an interesting feature to discriminate "straight-ahead" rock music from rhythmically complex world music, or classical music where the beat is not so accentuated.

Lambrou and Sandler (1998) computes "second order statistics" (angular second moment, correlation, entropy), which – we assume – similarly account for time structure in the data (although he does not motivate nor explain their use).

Soltau (1998) uses a very specific set of features, computed from the signal by a neural network in an unsupervised way. In a nutshell, it learns a set of abstract musical events from the tune (like elementary "notes," although Soltau's events are abstract, obtained from the hidden layer of a neural network, and represent patterns in the time-frequency representation which may lack any musical meaning or symbolic translation). The input vector for the classifier is a vector of statistical measures on these abstract events: co-occurrence of event i and j, etc. Thus, the system extracts information about the temporal structure of the songs. This coding may not be interpreted by human beings, but it is hopefully helpful for a machine learning algorithm to distinguish between genres.

Pitch related

To our knowledge, there has been only one explicit attempt at building a genre classifier for audio signals based on pitch features. Ermolinskiy et al. (2001) use pitch histograms feature vectors (either computed from audio signals or directly derived from MIDI data). The histograms can be folded into one single octave, which yields a representation comparable to Wakefield's chromogram (Wakefield, 1999) describing the harmonic content of the music, or can be left unfolded, which is useful for determining the pitch range of the piece. In Ermolinskiy et al. (2001) the histograms are compared and some patterns are found which reveal genre-specific information: rock songs typically have a small number of well-pronounced histogram peaks, because "they seldom exhibit a high degree of harmonic variation." On the other hand, jazz songs have more dense histograms, as "most notes between C0 and C5 are played at least once."

However, we have to mention here the many works concerning the related problem of "performance style recognition," i.e., recognizing if a jazz solo is rather like Charlie Parker's or John Coltrane's, or if this piece of piano music is played in a lyrical, pointillist, etc. style (Dannenberg et al., 1997; Chai & Vercoe, 2001). All this works typically deal with symbolic MIDI data, where the pitch information is given.

3.2 Classification

Supervised learning

All the Machine-Learning algorithms used in the prescriptive approach are supervised: In a first stage -training-, models of a few musical genres are built with some manually labeled data (i.e., a certain number of music files given with their genre). In a second stage -recognition-, these models are used to classify unlabelled data.

Simple taxonomies, with inconsistencies

The taxonomy of musical genre used in this supervised approach (i.e., the number of models that are built in the training stage) is always very simple and incomplete, so the resulting approach looks more like a proof of concept than a complete useful labelling system. There are usually very few classes:

– 3 in Tzanetakis and Cook (2000a); Lambrou and Sandler (1998) and Deshpande et al. (2001): "Classical, Modern, & Jazz," and "Rock, Piano & Jazz."
– 4 in Talupur et al. (2000) and Soltau (1998): "Classical, Jazz, Rock & Country," and "Classical, Rock, Pop & Techno."
– 5 in Ermolinskiy et al. (2001): "Electronic, Classical, Jazz, Irish Folk, & Rock."
– 6 in Tzanetakis et al. (2001) and Pye (2000): "Classical, Country, Disco, HipHop, Jazz & Rock" and "Blues, Easy Listening, Classical, Opera, Dance(Techno) & Indie Rock."

Some authors also use a "garbage music" general category to identify music outside the classification, although the results and consequences of this are not discussed.

Note that even with this simplified and incomplete framework, there are many ambiguities and inconsistencies in the chosen taxonomies:

– In "Rock, Piano, Jazz," where should a solo by Bill Evans be classified?
– What is the difference between "Classical" and "Opera"?

– Why classify "Indie Rock" and not "Rock" in the first place?
– What exactly is "Modern" compared to "Jazz"?

This illustrates what was already shown in section 2.1. for the general case of more exhaustive classifications: music genre taxonomies are highly inconsistent and dependent on users. One can question the soundness of building-in such ill-defined classes in an automatic system in the first place.

Static modeling against time-modeling

To our knowledge, only one system aims at an explicit time-modeling of music genre: Soltau (1998) compares a HMM with cepstral coefficients as input, and his above-stated explicit time features with a static NN classifier. All the other systems compute features on a frame-by-frame basis, and classify each frame separately. The classification scores are then summed over all frames, so that a global score for the whole song can be produced. No relationship between frames of the same song is taken into account in the classifier. This is surprising, especially because time-modeling (notably HMMs) has proved to be a successful framework for the related problem of "style recognition," mentioned above (see notably recent work by Pachet, 2002).

(static) Machine learning algorithms

There are three types of Machine Learning Algorithms used to classify the songs (or rather each of the feature vectors, as stated above):

– *Gaussian and Gaussian Mixture Models*, used in Tzanetakis and Cook (2000a); Tzanetakis et al. (2001); Pye (2000):
  This is used to estimate explicitly the probability density of each genre class over the feature space. The probability density is expressed as the weighted sum of simpler Gaussian densities, called components or states of the mixture. An equivalent definition is hierarchical sampling: to sample from the density, first draw a state at random (using a distribution over states) and then sample from that component. Most of the time, the estimation of the Gaussian parameters (mean and covariance) is done by Expectation-Maximization (EM) (Bishop, 1995).
– *Linear or non-linear classifier*, used in Talupur et al. (2000); Soltau (1998):
  This is often implemented by a feed-forward neural network, which learns a mapping between the high dimensional space of the feature vectors onto the different classes. The data may be first subjected to a nonlinear transformation before feeding it to a linear classifier. The new features are like basis functions in basis function regression, and the classifier is essentially thresholding the basis function regression (see for instance Bishop (1995)).
– *Vector Quantization*, used in Talupur et al. (2000); Pye (2000); Deshpande et al. (2001):
  Training a vector Quantizer is a way to find a good set of reference vectors (a code book), which can quantify the whole feature set with little distortion. Talupur et al. (2000) use a Kohonen Self-Organizing Map that both learns codevectors, and classifies the test data (with a nearest neighbor algorithm). Pye (2000) uses a tree-based vector quantizer, where a tree forms histogram templates for each genre, which are matched to the histograms of the test data.

Classification results and evaluation

An exhaustive comparison of these approaches is unfortunately impossible to perform since the authors choose different target taxonomies, different training sets, and, implicitly, different definition of "genre." However, we summarize here the most interesting results and remarks about these works.

– 48% of successful classification in Ermolinskiy et al. (2001) using 100 songs for each class in the training phase. This result has to be taken with care since the system uses only pitch information.
– Tzanetakis et al. (2001) achieves a rather disappointing 57%, but also reports 75% in Tzanetakis and Cook (2000a) using 50 songs per class.
– 90% in Lambrou and Sandler (1998) and 75% in Deshpande et al. (2001) on a very small training and test set, which may not be representative.
– Pye (2000) reports 90% on a total set of 175 songs.
– Soltau (1998) reports 80% with HMM, 86% with NN, with a database of 360 songs.

Some authors study genre specific errors (in Talupur et al. (2000): Classical = 95% success, Jazz = 85%, Rock = 80%, Country = 70%), and confusion matrices between genre (in Soltau (1998): 27.3% of "pop" is mistaken for "rock"). A common remark is that "classical music" and "techno" is easy to classify, while "rock," "country" and "pop" are not. A possible explanation for this is that the global frequency distribution of classical music is very different from techno (notably, the latter has strong bass frequencies), whereas many pop and rock songs use the same instrumentation. This suggests that timbre is not necessarily a good criterion to re-build an arbitrary genre taxonomy. This argument will be developed in section 3.3.

Perceptual study

Soltau (1998) has an interesting argument on the similarity of pop and rock genre: he has led a Turing Test on a group of 37 subjects, exposed to the same samples used for the machine learning test set: human confusions in the experiment are similar to confusions of the automatic system (complete results of this study are in Soltau's Diplomearbeit, only available in German). Note that similar conclusions are drawn by Pachet and Cazaly (2000): genre taxonomies are

highly non-consistent, and there is no general consensus on genre, especially for this pop/rock distinction.

Different genres have different and non-overlapping classification criteria

Talupur et al. (2000) lead an interesting "sensitivity analysis": which part of their 128-long input vectors (reminder: FFT coeff) contribute max in prediction? Notably, for Jazz, the learning is much better when a subset of the 64 elements is used (from 85% success to 91% when using only the second half of the input vectors). They conclude that the most important part of the spectrum for genre classification is the second quarter (although not specifying the frequency range more precisely), and then "hardest to learn" is the last quarter (high frequencies). More important is that this suggests that a unique set of features describing timbre is not optimal to classify different genres: jazz is very well classified when looking only at high frequencies, while techno needs looking only at bass frequencies (and gets very confused with rock, say, if high frequencies are included in the feature set).

3.3 Comments and suggestions

In a nutshell, most systems rely on the extraction of low-level descriptors of "global" timbre, and frame-by-frame classification of these features into a built-in taxonomy of genre (the taxonomy is built-in in the sense that a model first has to be trained for each genre we want to classify). This approach leads to 3 main problems.

Genre dependent features

As suggested in (Talupur et al., 2000) a unique set of features is not optimal to classify different genres: different genres have different classification criteria. For instance, "HipHop" would use bass frequency, drum track or voice timbre, while "Classical" would use medium frequencies (violin timbre, etc.). Moreover, the criteria are non-overlapping, since maybe "HipHop" and "Rock" are similar when looking at a certain frequency range, which is detrimental to a good classification.

This is a problem since:

- Feature selection is still a hard issue to be done automatically.
- Defining the best feature set is obviously data dependent: adding new titles in the database or new genre classes to the classifier will modify the optimal feature set, and moreover there is no guarantee that a given training set of sound samples is representative enough of the whole space in order to select the right feature set.

Taxonomic problems

Apart from the inconsistency inherent to any genre taxonomy, taxonomic issues include:

- *Need for hierarchical classification*: Weare in Dannenberg et al. (2001) suggests that an ideal system should only allow "graceful errors": "*if the [genre] is wrong, it is not so bad if an* '*East Coast Rap*' *song is classified as* '*Southern Rap*' *song, but if that song is mistakenly classified as* '*Baroque*' *[then] the error is quite painful.*" In other words, misclassification should occur between sub-genres of the same category ("Rap"). This suggests an incremental hierarchical classification (first classify into genres, then into subgenres, etc.), which cannot be done easily with low-level features. This would be easier with high-level features used in a kind of event-scenario: "is there guitar?", "is the drum-track using a lot of high-hat?" . . . But this is only wishful thinking, as the automatic extraction of such high-level features in polyphonic recordings is still not state-of-art.
- *Growth*: Tzanetakis et al. (2001) state the need to expand the genre hierarchy used in automatic systems both in width (new genres) and depth (new sub-genres). This also is problematic with the usual approach: for each new class (genre or sub-genre) that is added into the taxonomy, we have to create a new model with a new training stage (and maybe a new adapted feature set in the scope of automatic feature selection, which would require re-training all models of all genres. . . .). An ideal system would be more evolutive, thus allowing the user to define her own genres, and to add newly emerging musical genres (e.g., the newly advertised genre of "electronic pop"). Also a problem is the size and precision of the ideal taxonomy: Currently, there exist over a thousand musical genre categories in the MSN Music Search Engine (Dannenberg et al., 2001). If current results already have difficulties to classify "Rock" and "Pop", we hardly see how it can succeed in discriminating finer subgenres of the same categories ("Indie Rock", "Surf Rock" . . .).

Genre is being classified with intrinsic attributes

The systematic use of low-level features comes from a general faith in what's called "musical surface." A widely quoted study by Perrot and Gjerdigen (1999) (quoted in Tzanetakis et al. (2001) and Scheirer (2000)) shows that human can accurately predict whether they like a musical piece or not based on only 250 milliseconds of audio (while scanning the radio, for instance). This would suggest that humans can judge genre by using only an immediately accessible "surface" (a kind of timbre + texture + instrumentation + . . . ?) and without constructing any higher level theoretical description (such as "is there guitar?", "is the tempo fast or slow?" . . .). Moreover, automatic approaches from the signal assume that genre can be extracted from intrinsic attributes of the audio signal, which seems contradictory with the fact that in many cases genre is an extrinsic, emerging property.

Indeed, given an arbitrary genre taxonomy, there are very many counter-examples that one can find where pieces of two different genres have very similar "timbre":

Table 1. Average number of closest songs with the same genre as the query.

Number of Timbre Neighbors	Average number of songs in the same genre
Closest 1	0.43
Closest 5	1.43
Closest 10	2.56
Closest 20	4.61
Closest 100	18.09

Table 2. Measures of the overlap between different genres.

Average distance between titles	27.15
Average distance between titles of the same genre	26.91
Average distance between titles of different genres	27.17
Overlap on same genre	57.1%
Overlap on different genre	27.1%
Precision	14.1%
Recall	61.2%

– a Schumann sonata ("Classical") and a Bill Evans piece ("Jazz")
– a Prokoviev Symphony ("Classical") and an orchestral rendering of Gershwin's Porgy and Bess ("Jazz"), . . .

Similarly, there are many examples of pieces of the same genre that have very different timbre:

– two songs by The Beatles : "*Helter Skelter*" (heavy overloaded guitars), and "*Lucy in the Sky*" (strange harpsichord-like)
– two jazz pieces: "*A Love Supreme*" by John Coltrane, and "*My Funny Valentine*" sung by Chet Baker, . . .

We have led a quantitative study of the correlation between timbre similarity and genre, using the 20000 music titles Cuidado database. As described in section 2.2., each title in the database has been manually labeled in the Cuidado genre taxonomy. Timbral similarity between titles is estimated using MFCC and Gaussian Mixture Models, as described in Aucouturier (2002). This technique, used for classification by Pye (2000), was adopted because it yields the best classification results among all the algorithms reviewed earlier (3.2.5.). For each title in the database, we compute its "timbral distance" to all the other titles, and compare these distances to the genre of the titles (only the root level of the Cuidado genre taxonomy is used, i.e., 18 genre families: Ambiance, Blues, Classical, Country, Electronica, Folk, Hard, Hip Hop, Jazz, New Age, Pop, Reggae, Rhythm&Blues, Rock, Rock&Roll, Soul, Variety, World).

Results can be seen in Table 1 and Table 2. Both tables show that for a given taxonomy, there is a very poor correlation between genre and timbre. In an Information Retrieval point of view, the precision of a query on genre based on timbral distance is very low (15%). In Table 2, "Overlap on Same Genre" is the ratio

$$\frac{N_{diff<same}}{N_{diff}}$$

where N_{diff} is the total number of songs with a different genre as the query's, and $N_{diff<same}$ is the number of songs in N_{diff} whose timbral distance to the query is smaller than the mean distance to songs of the same genre. Similarly, "Overlap on Different Genre" describes the proportion of songs which have the same genre as the query, but whose distance to the query is larger than the mean distance to songs of the different genre. Both values are high, which suggests that prescriptive classification schemes based on timbre are intrinsically limited, and cannot scale in both in the number of music titles, and in the number of genre classes.

4. Emerging genre classifications from similarity relations

The second approach to automatic genre classification is exactly opposite to the *prescriptive* approach just reviewed. Instead of assuming that a genre taxonomy is given a priori, it tries to emerge a classification from the database, by clustering songs according to a given measure of similarity. While the prescriptive approach adopts the framework of supervised learning, this second point-of-view is unsupervised. Another important difference is that in the first approach, genre classifications are considered as natural and objective (we have seen problems about this in section 2.1.), whereas in this approach it is similarity relations which are considered as objective.

4.1 Measures of similarity

– *Intrinsic attributes from the signal*: The same features that were described in section 3.1. can be used to assess similarity between individual titles. This was notably done by the authors in the study described in section 3.3. Similar distance functions are used for instance in the "Muscle Fish" technology (Wold & Blum, 1996), or by Foote (1997). However, these works about audio similarity are on the edge of genre classification, and have never been applied to genre in an explicit manner. Therefore, they are not to be reviewed here. Besides, we have already made arguments that intrinsic signal attributes are not always correlated with what is usually described as genre.
– *Cultural similarity from text documents*: In the rest of this review, we describe a music similarity measure based on data-mining techniques that appear to be well suited to genre clustering. These techniques are able to extract similarities that are not possible to extract from the audio signal.

4.2 Collaborative filtering

Collaborative Filtering (CF) (Shardanand & Maes, 1995) is based on the idea that there are patterns in tastes: tastes are not distributed uniformly. These patterns can be exploited very simply by managing a profile for each user connected to the service. The profile is typically a set of associations of items to grades. In the recommendation phase, the system looks for all the agents having a similar profile than the user's. It then looks for items liked by these similar agents which are not known by the user, and finally recommends these items to him/her.

Experimental results show that the recommendations, at least for simple profiles, are of good quality, once a sufficient amount of initial ratings is given by the user (Shardanand & Maes, 1995). However, there are limitations to this approach, which appear by studying quantitative simulations of CF systems (Epstein, 1996). The first one is the inclination to "cluster formation," which is induced by the very dynamics of the system. CF systems produce interesting recommendations for naive profiles, but get stuck when the profiles get bigger: eclectic profiles are disadvantaged.

Another problem, shown experimentally, is that the dynamics favors the creation of hits, i.e., items which are liked by a huge fraction of the population. If hits are not a bad thing in themselves, they nevertheless limit the possibility of other items to "survive" in a world dominated by weight sums.

CF has been used with some success in the field of music selection (Pestoni et al., 2001; French & Hauver, 2001). However, it has a number of issues for EMD systems: as put in Pye (2000), it *"require[s] considerable data and [is] only applicable [for new titles] some time after [their] release."* Also a problem is that it is difficult to guarantee that the extracted taste/buying patterns are linked to a genre similarity. Finally, cluster formation and uneven distribution of chances for items (e.g., hits) are important drawbacks of the approach, both from the user viewpoint (clusters from which it is difficult to escape), and the content provider viewpoint (no systematic exploitation of the catalogue).

4.3 Co-occurrence analysis

We have introduced co-occurrence techniques to automatically extract musical similarity between titles or between artists (Pachet et al., 2001). The technique yields a distance matrix for arbitrary sets of items. It was applied to two different music sources, and experiments were conducted on various title and artist databases.

Sources

We have investigated two possible sources: radio programs, and databases of compilation CDs.

– *Radio programs*: The rationale behind analyzing radio programs is that usually, at least for certain radio stations, the choice of the titles played and the choice of their sequence is not arbitrary. The radio programmer has, in general, a vast knowledge of the music he or she plays on air, and this knowledge is precisely what gives the program its characteristic touch. For instance, some radio stations specialize in back catalogues of the sixties (in France e.g., Radio Nostalgie and Europe 2), others in non-contemporary classical music (Radio Classique), and yet others have more diverse catalogues (such as FIP/Radio France). In all cases, however, the titles and their sequencing are carefully selected in order to avoid breaking the identity of the program. It is this very knowledge (choice of titles and choice of sequencing) that we wish to utilize by data mining. Several thousands radio stations exist in the occidental world, and many of them make their programs available on the web, or through various central organizations, such as Broadcast Data Systems. For our experiments we have chosen a French radio station that has the advantage of not being specialized in a particular music genre: Fip (Radio France).
– *Track Listing Databases*: Another important source of information is actual CD albums, and in particular, samplers (compilations). Compilations, either official ones produced by labels, or those made by individuals, often carry some overall consistency. For instance, titles on compilations such as "Best of Italian Love Songs," "French Baroque Music," or "Hits of 1984" have explicit similarities of various sorts (here, social impact, genre, and period). Our main hypothesis is that if two titles co-occur in different compilations, this reinforces the evidence of some form of similarity between them.

Co-occurrence techniques

Co-occurrence analysis consists in building a matrix with all titles in row and in column. The value at (i, j) corresponds to the number of times that titles i and j appeared together, either on the same sampler, on the same web page, or as neighbors in a given radio program.

To define an actual distance function, we need to take into account several important factors. First, two titles may never co-occur directly, but they may each co-occur with a third title. The distance function should take such indirect co-occurrence into account. Second, because we want to assess both the soundness (all found similarities are "good") and completeness (all "good" similarities are found) of the extracted similarities, we need to restrict the validation to a close corpus of titles that can then be used for comparisons with human similarity judgments.

Given a corpus of titles $S = (T_1, \ldots, T_N)$, we compute the co-occurrence between all pairs of titles T_i and T_j. The co-occurrence of T_i with itself is simply the number of occurrences of T_i in the considered corpus. Each title is thus represented as a vector, with the components of the vector being the co-occurrence counts with the other titles.

Components of each vector are normalized to eliminate frequency effects of the titles.

Results

These experiments show that the technique is indeed able to extract similarities between items. We have determined that 70% of the clusters constructed from such data mining distances translate interesting similarities. Specific music genres are quite well distinguished: for instance in our experiments, two jazz guitar titles (Wes Montgomery – Midnight Mood, and Jim Hall – Body and Soul) are clustered together. A complete description of the results so far can be found in Pachet et al. (2001).

Issues

– *The clusters are not labeled*: Characterizing the nature of the extracted similarities is not easy. Besides common artist similarities, two main kinds of similarity relations for CDDB were identified: thematic/genre similarity, and similarity of period (coming probably from the abundance of "best of the year" samplers). For the radio (FIP), the similarity relations are quite different. Current experiments on a database of 5000 titles show that artist consistency is not enforced as systematically as in the other data sources. Moreover, the similarities are more metaphorical, and in some sense less obvious, therefore often more interesting. They can be of various kinds:
  – covers, e.g., "*Lady Madonna*" by the Baroque Ensemble is close to "*Ticket to Ride*" by the Beatles,
  – instrument/orchestration, e.g., *Eleanor Rigby* and a Haydn quartet,
  – based on title names or actual meaning of the lyrics, e.g., "*Kiss* – Prince" close to "Le *Baiser* – Alain Souchon." Therefore, it is often not clear whether the extracted similarity is an indication of genre, or rather timbre, semantics . . . One possible cure for this is to select sources that are likely to be genre specific (e.g., the "Best of Italian Love Songs" sampler, or playlists from a jazz radio, etc.).
– *Works only for titles appearing in the sources*: One obvious drawback is that this technique only works for titles appearing in the analyzed sources. This is a problem in the context of an EMD system: similarities can only be computed for a subset of the titles – or rather part of the possible duplets (T_i, T_j) of titles – in a database or a catalogue. This suggests that this technique should be used in conjunction with the other sources of similarities described in this article.

5. Conclusion

We have described three approaches for extracting musical genre.

– A manual classification of titles can be useful for bootstrapping and evaluating automatic systems, but it is not realistic for large databases, and does not easily scale-up. We have chosen to classify artists, which gives a good trade-off between precision and scalability. In any case, analysis of existing genre taxonomies show that genre is an ill-defined notion.
– We have reviewed signal techniques, and shown that they mostly rely on supervised classification techniques on a space of low-level timbral features. Success is limited to small taxonomies (distinction between 1–5 families), and very distinct genres (classical music and techno). This prescriptive approach is limited by the inconsistencies of the built-in taxonomy, and the assumption that genre can be assessed from intrinsic signal attributes. We have shown that for a given genre taxonomy, correlation between genre classes and timbre similarity can be very poor.
– Data mining techniques such as co-occurrence analysis are able to extract high-level similarities between titles and artists, and are therefore well-suited to the unsupervised clustering of songs into meaningful genre-like categories. These techniques suffer from technical problems, such as the labeling of clusters, but these issues are currently under study and better schemes should be devised soon.

The approaches presented here differ in the techniques used, but more importantly in the implicit conception of genre they are based on. Prescriptive approaches consider genre as an object of study, while emerging approaches attempt to construct operational definitions of genre.

Acknowledgement

This work has been founded by the European IST project CUIDADO (Content-based Unified Interfaces and Descriptors for Audio/music Databases available On-line).

References

Aucouturier, J.-J., & Pachet, F. (2002). Music Similarity Measures: What's the Use? In *Proceedings of the 3rd International Symposium on Music Information Retrieval (ISMIR) 2002*.

Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.

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

Dannenberg, R. et al. (2001). Panel: New Directions in music information retrieval. In *Proceedings of the 2001 International Computer Music Conference*.

Dannenberg, R.B., Thom, B., & Watson, D. (1997). A Machine Learning Approach to Musical Style Recognition. In: *Proceedings of the 1997 International Computer Music Conference*.

Deshpande, H., Nam, U., & Singh, R. (2001). Classification of music signals in the visual domain. In *Proceedings of the 2001 Digital Audio Effects Workshop*.

Epstein, J. (1996). Growing Artificial Societies: Social Science from the Bottom Up. Cambridge: *MIT Press*.

Ermolinskiy, A., Cook, P., & Tzanetakis, G. (2001). Musical Genre classification based on the analysis of harmonic content features in audio and midi. *Work Report,* Princeton University.

Eronen, A., & Klapuri, A. (2000). Musical instrument recognition using cepstral coefficients and temporal features. In: *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP) 2000*.

Foote, J.T. (1997). Content-based retrieval of music and audio. *Multimedia Storage and Archiving Systems II, Proc. of SPIE*, *3229*, 138–147.

French, J.C., & Hauver, D.B. (2001). Flycasting: On the fly broadcasting. In *Proceedings of the WedelMusic Conference*. Rome, Italy.

ISO/IEC International Standard IS 11172–3 "Information Technology – Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s Part 3: Audio."

Lambrou, T., Kudumakis, P., Sandler, M., Speller, R., & Linney, A. (1998). Classification of Audio Signals using Statistical Features on Time and Wavelet Transform Domains. In *Proc 1998 IEEE International Conference on Acoustic Speech and Signal Processing (ICASSP)*.

Martin, K. (1998). Towards automatic sound source recognition: Identifying Musical Instruments. *In Proceedings NATO Computational Hearing Advanced Study Institute*. Rome, Italy.

Pachet, F., & Cazaly, D. (2000). A taxonomy of musical genres. In *Proc. Content-Based Multimedia Information Access (RIAO)*, Paris, France.

Pachet, F. (2001). Metadata for music and sounds: The Cuidado project. In *Proceedings of the Content-Based Multimedia Indexing (CBMI) Workshop*, *University of Brescia, September*, *2001*.

Pachet, F. (2002). The Continuator: Musical Interaction With Style. In *Proceedings of the 2002 International Computer Music Conference (ICMC)*.

Pachet, F., Westermann, G., & Laigre, D. (2001). Musical Datamining for EMD. In *Proceedings WedelMusic Conference*. Italy.

Perrot, D., & Gjerdigen, R.O. (1999). Scanning the dial: An exploration of factors in the identification of musical style. *In Proceedings of the 1999 Society for Music Perception and Cognition pp. 88 (abstract)*.

Pestoni, F., Wolf, J., Habib, A., & Mueller, A. (2001). KARC: Radio Research. In *Proceedings WedelMusic Conference*. Italy, 2001.

Pye, D. (2000). Content-Based Methods for Managing Electronic Music. In *Proc 2000 IEEE International Conference on Acoustic Speech and Signal Processing (ICASSP)*.

Rabiner, L.R., & Juang, B.H. (1993). Fundamentals of speech recognition. NJ: Prentice-Hall.

Scheirer, E. (2000). Music Listening Systems. *PhD Thesis.* MIT MediaLab Cambridge, MA, *USA*.

Shardanand, U., & Maes, P. (1995). Social information filtering: Algorithms for automating word of mouth. In *Proceedings ACM Conference on Human Factors in Computing Systems*.

Soltau, H. (1998). Recognition of Musical Types. In *Proceedings International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Talupur, M., Nath, S., & Yan, H. (2000). Classification of Musical Genre. *Working Paper, Computer Science Department. Carnegie Mellon University*.

Tzanetakis, G., & Cook, P. (2000a). Audio Information Retrieval (air) Tools. In *Proceedings of the 2000 International Symposium on Music Information Retrieval*.

Tzanetakis, G., & Cook, P. (2000b). Sound Analysis Using Mpeg Compressed Audio. In: *Proceedings ICASSP 2000*.

Tzanetakis, G., Essl, G., & Cook, P. (2001). Automatic Musical Genre Classification of Audio Signals. In *Proceedings of the 2001 International Symposium on Music Information Retrieval*.

Wakefield, G.H. (1999). Mathematical Representation of Joint Time-Chroma Distributions. *In Proceedings SPIE International Symposium on Optical Science, Engineering and Instrumentation, Denver, Colorado*.

Wold, E., & Blum, T. (1996). Content based classification, search and retrieval of audio. *IEEE Multimedia*, 3(3).