EXTRACTING AUTOMATICALLY THE PERCEIVED INTENSITY OF MUSIC TITLES

Aymeric Zils, François Pachet

Sony CSL Paris
{zils, pachet}@csl.sony.fr

ABSTRACT

We address the issue of extracting automatically high-level musical descriptors out of their raw audio signal. This work focuses on the extraction of the perceived intensity of music titles, that evaluates how energic the music is perceived by listeners.
We present here first the perceptive tests that we have conducted, in order to evaluate the relevance and the universality of the perceived intensity descriptor. Then we present several methods used to extract relevant features used to build automatic intensity extractors: usual Mpeg7 low level features, empirical method, and features automatically found using our Extractor Discovery System (EDS), and compare the final performances of their extractors.

1. INTRODUCTION

The exploding field of Music Information Retrieval has recently created extra pressure to the community of audio signal processing, for extracting automatically high-level music descriptors. Indeed, current systems propose users with millions of music titles (e.g. the peer-to-peer systems such as Kazaa) and query functions limited usually to string matching on title names. The natural extension of these systems is content-based access, i.e. the possibility to access music titles based on their actual content, rather than on file names. Existing systems today are mostly based on editorial information (e.g. Kazaa), or metadata which is entered manually, either by pools of experts (e.g. All Music Guide) or in a collaborative manner (e.g. the MoodLogic). Because these methods are costly and do not allow scale up, the issue of extracting automatically high-level features from the acoustic signals is key to the success of online music access systems.

Extracting automatically content from music titles is a long story. Many attempts have been made to identify dimensions of music that are perceptually relevant and can be extracted automatically. One of the most known is tempo or beat, that is a very important dimension of music that makes sense to any listener. However, there are many other dimensions of music that are perceptually relevant, and that could be extracted from the signal. For instance, the presence of voice in a music title, i.e. the distinction between instrumentals and songs is an important characteristic of a title. We focus here on another example: the *perceived intensity*. It makes sense to extract the subjective impression of energy that music titles convey, independently of the RMS volume level: with the same volume, a Hard-rock music title conveys more energy than, says, an acoustic guitar ballad with a soft voice.

Extracting a perceptive descriptor raises two main issues:

- First, we have to prove the relevance and the universality of the descriptor, by conducting perceptive tests on a set of listeners.

- Second, once the descriptor is proven relevant, the information has to be extracted automatically; the task is difficult because polyphonic music signals are usually highly complex in nature. We experiment here several methods to detect relevant characteristics of the audio signal, enabling us to evaluate the intensity of music titles.

2. PERCEPTIVE TESTS ON PERCEIVED INTENSITY

Musical intensity is a subjective descriptor, that every listener perceives differently. In order to build a global model of this descriptor, we have to find a consensual basis in the diverse perceptions of intensity, and prove that it is relevant to generalize our model. We conducted 2 series of perceptive tests; we present here the first tests that were evaluated on a database containing 204 musical signals of duration 10s, with a priori various intensities. These signal are then used as a learning database to build intensity extractors. The second tests were performed on another 200 signals database, used as a test database for our extractors.

2.1. Presentation of the tests

The tests were done on people from our lab, and were also accessible on the web. They consisted in listening to one of the 204 musical extracts, and evaluating its intensity level, by choosing a category among Low - Medium - High -Very High.

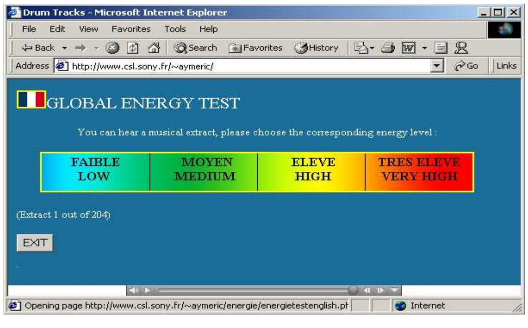

Fig. 1: *Web Page of the Perceived Intensity Tests*

2.2. Results

We obtained more than 2600 answers, that corresponds to approximately 12-13 answers for each title of the database.

Then a general perceived intensity value is computed for each title of the database, by removing the extreme results, assigning a numeric value for each energy level (normalized between 0 for 'Low' and 1 for 'Very High'), and taking the mean value on all listeners. To evaluate the relevance of this intensity value, we compute the standard deviation of the results for all the listeners: for 200 titles (98%), the standard deviation is less than the distance between two successive categories (0.33), so the mean value is assumed to be a correct evaluation of their intensity. The other titles are removed from the database, as we consider that their intensity is too subjective to be evaluated.

These final intensity values are then used as learning references to compute our model, and built automatic extractors. We proceed the same on the second series of perceptive tests, in order to build a 200 titles database, used to test these models and extractors.

3. EXTRACTING FEATURES FOR INTENSITY USING TRADITIONAL METHOD

We present here the different methods that were tested to find relevant extractors for the intensity of music titles: combination of low-level features, empirical method, and EDS, our system using a genetic algorithm to build relevant features for given problems. The features are evaluated by computing the correlation between the function values and the perceptive values, by cross-validation on the 200 remaining titles of the learning database.
As a comparison, note that the correlation of a random function is 0.18.

3.1. Usual Mpeg7 Audio Features

We first used a traditional approach to build high-level music descriptors, that consists in combining selected features out of a set of lower-level ones (for example [1]).Since there is no signal processing state of the art in evaluating the perceived intensity, we filled the set with audio descriptors that are known to be relevant for audio description problems, such as those described in Mpeg7 (see [2]). We tested 30 Mpeg7-like features, based on temporal and spectral features among which amplitudes, energies, high-frequency content, spectral flatness, spectral centroïd, and so on...
The most correlated feature is the *spectral skewness*, with a correlation of 0.56, which provided an model error of 16.9% compared to the results of the perceptive tests. The most relevant combination of these features has a correlation of 0.87, with a model error of 12.1%.

3.2. Empirical Method

We had different intuitions on the origin of the intensity of music. The two main intuitions were that the intensity is simply linked to the tempo, or more complexly to the variations of energy of the signal. Several new features were built out of these intuitions: for instance, we used Scheirer's tempo extractor [3], and studied different signal processing functions describing the signal energy variations. We evaluated the correlation of all these features with the perceptive tests. The most correlated function that we found is:

Log (Variance (Derivation (Energy (Audio Signal))))

with a correlation of 0.57. This function is linked with the amplitude and the frequency of the variations of the signal's raw energy. The tempo, automatically extracted with Eric Scheirer's method, had a correlation of 0.55.

4. EXTRACTING FEATURES FOR INTENSITY WITH THE EDS SYSTEM

EDS (Extractor Discovery System), developed at Sony CSL, is a heuristic-based generic approach for extracting automatically high-level music descriptors from acoustic signals. EDS approach is based on Genetic Programming (see [4]), used to build extraction functions as compositions of basic mathematical and signal processing operators.

4.1. Presentation of EDS

4.1.1. Global architecture

Considering:
- A given description problem: classification or regression (here the evaluation of the global intensity),
- A database of audio signals with the associated perceptive values (normalized intensity here).

EDS consists in 2 steps: (1) genetic search algorithm builds relevant signal processing features for the description problem, and (2) machine learning algorithms build the associated extractors from these features. The global architecture of the system is presented in Fig. 2:

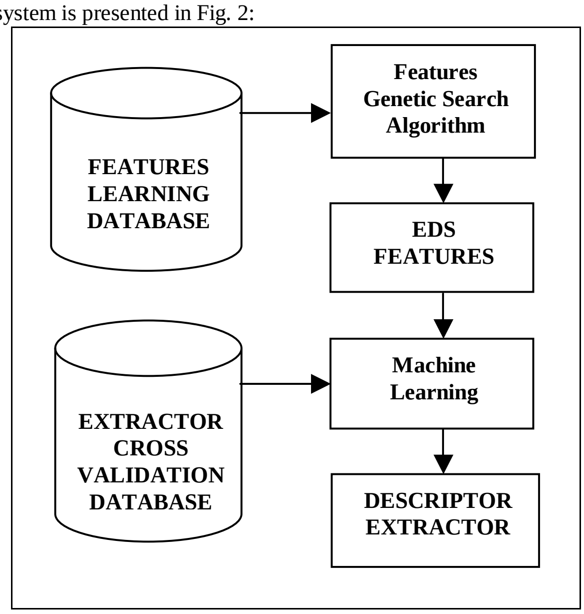

Fig. 2: *EDS Global Architecture*

In the features genetic search phase, the search is guided by specialized heuristics that embody knowledge about the signal processing functions built by the system. Signal processing patterns are used in order to control the general function extraction methods. Rewriting rules are introduced to simplify overly complex expressions. In addition, a caching system further reduces the computing cost of each cycle.

4.1.2. EDS functions

EDS builds functions as compositions of signal processing and mathematical operators, such as 'MEAN (X)', that takes the mean value of the set X, or 'HPFILTER (X, Fc)', that filters the signal X with a cut-off frequency Fc. For instance:

MEAN (MAX (FTT (SPLIT (HPFILER (Signal, 1000Hz), 10ms)))

- high-pass filters the audio signal at 1000Hz,
- then splits the resulting signal into 10ms frames,
- then takes MAX of the FFT of each frame,
- and finally provides the MEAN value (on all the frames)

4.1.3. EDS data types

Typing rules allow to control the input and output types for each operator, and consequently the syntax of the global function. The need for typing is well-known in Genetic Programming, to ensure that the functions generated are at least syntactically correct. Different type systems have been proposed for GP, such as strong typing ([5), that mainly differentiate between the “programming” types of the inputs and outputs of functions.

In EDS, we distinguish data at the level of their “physical dimension”: For instance, audio signals and spectrum are both vectors of floats, but are different in their dimensions: a signal is a time to amplitude representation, while a spectrum associates frequency to amplitude. Thus, we have 3 "physical" atomics types: time “t”, frequency "f", and amplitudes “a”, that allows to build more complex types, such as functions (representations from one dimension to another, for example the type of an audio signal [time to amplitude representation] is "t:a"), and vectors (special cases of functions associating an index to a value, for example, a list of time onsets in an audio signal is notated "Vt").

4.1.4. EDS patterns

This typing system allows to build "generic operators" that stand for one or several random operator(s) whose output type and possible arguments are forced. 3 different generic operator (notated "*", "!", and "?") stand for 1 or several operator of given output types, with 1 or several given arguments.

These generic operators allow to write functions patterns, that stand for any function satisfying a given signal processing method. For instance, the pattern:

?_a (!_Va (Split (*_t:a (SIGNAL))))

stands for:

« Apply signal transformations in the temporal domain » (*_t:a)
« Split the resulting signal into frames » (Split)
« Find 1 characteristic value for each frame » (!_Va)
« Find 1 relevant characteristic value for the entire signal » (?_a)
This general extraction scheme can be instantiated as:
Sum (Square (Mean (Split (HpFilter (SIGNAL, 1000Hz), 100))))
These patterns can be specified in EDS to guide the search.

4.1.5. EDS heuristics

Heuristics are vital ingredients to guide the search and a central point in the design of EDS. They represent the know-how of signal processing experts, about functions seen a priori, i.e. before their evaluation. The interest of heuristics is that they both favor a priori interesting functions, and rule out obviously non-interesting ones.

A heuristic in EDS associates a score between 0 and 10 to a potential composition of operators, used to select candidates at all the function creation stages during the search. In addition, heuristics are useful to control the structure of the functions (such as the size of functions), avoid useless combinations of operators (such as redundancies), range constant values, etc...

4.1.6. EDS genetic search of functions

The system uses a genetic algorithm to build relevant features, that works as follows:

- EDS starts with building a random population of functions out of a set of operators. This creation is done in respect with the data types and the heuristics.
- computes the instantiations of the created functions with all the signals in the genetic search database
- evaluates the correlation of the values of the created functions with the values of the perceptive tests,
- selects the most correlated functions in the population,
- applies some genetic transformations on these functions, such as constants variations, mutations, replacements, and cross-over.
- creates a new population out of the transformed functions and new random functions.
- evaluates the new population is evaluated, and so on...

Theoretically, the system stops when a perfect function is found (correlation=1); practically we use the intermediate results (the best functions found since the beginning) as features to build models of intensity.

4.1.7. Genetic transformations of functions

Genetic transformations of functions in EDS are of different types: constants variations, mutations, replacements, cross-over.

Constants variation keeps the structure of the function but applies some slight variations on the constant values. For instance "Mean (HpFilter (Signal, **1500Hz**))" can be transformed into "Mean (HpFilter (Signal, **1400Hz**))".

Replacements replace 1 operator of the function by another operator that handles the same data types. For example, "Mean (**HpFilter** (Signal, 1500Hz))" can be transformed into "Mean (**LpFilter** (Signal, 1500Hz))".

Mutations replace 1 sequence of operators in the function by another sequence that handles the same data types. For example, "**Mean** (HpFilter (Signal, 1500Hz))" can be transformed into "**Max (Autocorrelation** (HpFilter (Signal, 1500Hz))))".

Cross-over replace 1 sequence of operators in a function by another sequence taken from another function, that handles the same data types. For example, a cross-over of "Mean (**HpFilter (Signal, 1500Hz)**)" and "**Max (Autocorrelation** (Signal)))" can be " **Max (Autocorrelation** (**HpFilter (Signal, 1500Hz)**)".

4.1.8. EDS optimization

2 optimizations speed-up EDS process: rewriting rules, and a system of caching of the most useful results.

Rewriting rules are applied to simplify functions before their evaluation, using a fixed point mechanism until to obtain a normal form. Unlike heuristics, they are not used by the genetic algorithm to favor combinations, but avoid computing several times the same function with different but equivalent forms, and reduce the computation cost (for example using Perseval equality avoids to compute the "Fourier transform" of a signal).

Finally, a caching mechanism is introduced, so that any costly function is computed once, and reused when possible: Every time a new function is computed, all the intermediate results are stored on separate files, and the caching technique consists in keeping in memory the most useful results, depending on their computation time, their utility, and their size.

4.2. Results of EDS

4.2.1. Features

We ran EDS on the perceived intensity problem. The search has provided automatically different types of relevant features.

First, EDS has found features close to Mpeg7 low-level descriptors, but improved with different pre-processings. For example, the most correlated function of this type found by the system is "*Mean (SpectralSkewness (Split (Signal, 0.1s)))*", that has a correlation of 0.60. This shows that EDS is able to improve automatically usual features by adding specific signal processing, which is usually done by hand by researchers.

Second, EDS has found features close to empirically found descriptors, but improved with additional operations. The most correlated of these functions is: "*Mean (Log (Variance (Split (Derivation (Square (Signal)), 1s))))*" with a correlation of 0.64.

Finally, EDS has found new features, such as "*Sqrt (Min (Sum (Fft (Split (Signal, 1s)))))*", that reaches a correlation of 0.69.

4.2.2. Extractor

Finally, we solve the regression problem of building the most efficient intensity extractor as possible, by computing the optimal combination of the most relevant functions found by the system. We obtain efficient extractors for audio signals of duration 10s, that is the length of audio signals in the database. The model error of our best extractor is 11.3%, compared to the perceptive results by cross-validation on our test database. It combines both Mpeg7 and the best features found by EDS.

To obtain a extractor for the global intensity of a whole song, whose duration is several minutes, we need to cut the signal into 10s frames, extract the local intensity on these frames, and then aggregate the local intensity values into one global value. For music browsing applications, we chose to provide the "mean intensity" of a song, by taking the mean value on all the frames. It is also interesting to draw a "song intensity profile" representing the successive intensity values along a song, in order to determine which parts of the song are more or less intense.

5. SUMMARY

Here is a table that summarizes the results of the intensity extractor for the different methods used:

METHOD	CORRELATION	MODEL ERROR
RANDOM (mean value for all titles)	0.18	21%
BEST MP7 Feature [Spectral Skewness (SIG)]	0.56	16.9% +- 1.5%
MP7 Features Combination 21 Selected Features	0.87	12.1% +- 1.9%
BEST EDS Feature [Sqrt(Min(Sum(Fft(Split (SIG,1s)))))]	0.68	14.5% +- 1.8%
EDS+MP7 Combination 18 Selected Features:	0.89	11.3% +- 1.8%

Fig. 3: *Performances of Intensity Extractors*

6. CONCLUSIONS

We presented several methods to extract relevant features concerning the problem of modelling the perceived energy of music titles out of their raw audio signal. We focused on the presentation of our EDS system. EDS is able to both improve usual low-level descriptors by adding pre- and post-processing operations, and to build automatically new relevant features, thanks to a genetic search algorithm guided by specialized heuristics. EDS is a general system that can be used to find relevant features for any description problem.

7. REFERENCES

[1] Tzanetakis George, Essl Georg, and Cook Perry, 'Automatic musical genre classification of audio signals', in Proceedings of 2nd International Symposium on Music Information Retrieval (ISMIR01), pages 205--210, Bloomington, IN, USA, October 2001

[2] Perfecto Herrera, Xavier Serra, Geoffroy Peeters, "Audio descriptors and descriptors schemes in the context of MPEG-7", Proceedings of the 1999 ICMC, Beijing, China, October 1999.

[3] Scheirer, Eric D., "Tempo and Beat Analysis of Acoustic Musical Signals", J. Acoust. Soc. Am. (JASA) 103:1 (Jan 1998), pp 588-601.

[4] John R. Koza, "Genetic Programming: on the programming of computers by means of natural selection", Cambridge, MA: The MIT Press.

[5] David J Montana, "Strongly typed genetic programming", In Evolutionary Computation 3-2, 1995, pp 199-230.