Pachet, F. Roy, P. Exploring Billions of audio features. CBMI 07, Bordeaux, 2007

EXPLORING BILLIONS OF AUDIO FEATURES

François Pachet, Pierre Roy

Sony CSL-Paris,
6, rue Amyot, 75005 Paris

ABSTRACT

Many works in audio and image signal analysis are based on the use of "features" to represent characteristics of sounds or images. Features are used in various ways, for instance as inputs to classifiers to categorize automatically objects, e.g. for audio scene description. Most, if not all, approaches focus on the development of clever classifiers and on the various processes of feature selection, classifier algorithms and parameter tuning. Strangely, the features themselves are rarely justified. The predominant paradigm consists in selecting, by hand, "generic", well-known features from the literature and focusing on the rest of the chain. In this study we try to generalize the notion of feature to make the choice of features more systematic and less prone to hazardous, unjustified human choices. We introduce to this aim the notion of "analytical feature": features built only from the analysis of the problem at hand, using a heuristic function generation process. We show some experiments aiming at answering some general questions about analytical features.

1 INTRODUCTION

Most audio classification approaches use either one of these two paradigms: 1) a general scheme, called *bag-of-frames* and 2) *ad hoc* approaches.

The bag-of-frame approach [1], [35] consists in considering the signal in a blind way, using a systematic and general scheme: the signal is sliced into consecutive, possibly overlapping frames (typically of 50ms), from which a vector of audio features is computed. The features are supposed to represent characteristic information of the signal for the problem at hand. These vectors are then aggregated (hence the "bag") and fed to the rest of the chain: First, a subset of available features is identified, using some feature selection algorithm. Then the feature set is used to train a classifier, from a database of labeled signals (training set). The classifier thus obtained is then usually tested against another database (test set) to assess its performance.

The use of the features as input to classifiers plays two roles: a dimension reduction role, and a representation role. Indeed, the signal itself could in principle be used as input to classifiers, but its dimension (number of samples) is usually too high with respect to the training set size, resulting in overfit. Additionally, the time/amplitude representation of signals has long been acknowledged to be poorly adapted to represent perceptive information: audio features used in the classification literature aim precisely at capturing essential perceptive characteristics of audio signals that are not obviously present in its temporal representation. A source of audio features is for instance MPEG7-audio [10] or more specifically [23] or [15] for the music domain. These features are usually of low dimensionality, and contain statistical information from the temporal domain (e.g. Zero-crossing rate), spectral domain (e.g. SpectralCentroid), or more perceptive aspects (such as sharpness, relative loudness, etc.).

The bag-of-frame approach has been used extensively in the MIR domain, for instance by [27]. A large proportion of MIR related papers has been devoted to studying the details of this chain of process: feature identification [23], feature aggregation [18]; feature selection [4],[17],[21], classifier comparison or tuning [1], [35].

An even larger proportion of Ismir papers (about 25% according to [1]) discuss the application of this approach to specific musical problems: genre classification [16], [20], [34], [33] ; orchestral sound classification [22], percussion instrument classification [32], [30] ; [31], [9], tabla strockes, [6], [3], noise classification [8] as well as identification tasks, such as vocal identification [12], mood detection [13].

This approach achieves indeed a reasonable degree of success on some problems. For instance, speech music discrimination systems based on the bag-of-frame paradigm yield almost perfect results. However, the approach shows limitations when applied to more "difficult" problems. Although classification difficulty is hard to define precisely, it can be noted that problems involving classes with a smaller degree of abstraction are usually much more difficult to solve. For instance, genre classification works well on abstract, large categories (Jazz vs Rock), but performance degrades for more precise classes (e.g. Be-bop versus Hard-bop).

In these cases, the natural tendency is usually to look for *ad hoc* approaches, which aim at extracting "manually" from the signal the characteristics most appropriate for the problem at hand, and exploit them accordingly. This can be done either by defining ad hoc features, integrated in the bag-of-frame approach (e.g. the 4-Hertz modulation energy used in some speech/music classifiers, [27], or by defining completely different schemes for classifying, e.g. the analysis-by-synthesis approach designed for drum sound classification [38], and further developed by [37] and [26].

One of the possible reasons for the limitation of the bag-of-frame approach is that the generic features used, such as the Mpeg-7 feature set, do not always capture the relevant perceptive characteristics of the signals to be classified. Some classifier algorithms, such as kernel methods [28] including Support Vector Machines [2], [29], do try to transform the feature space with the aim of improving inter-class separability. However, the increasing sophistication of feature selection or classifier algorithms cannot compensate for any lack of information in the initial features set.

Although ad hoc approaches may indeed reach interesting performance, they are rarely reusable: ad hoc features are, by definition specific to a problem. Consequently the scientific contribution (and epistemological status) of reports of ad hoc approaches is debatable.

In this work we try to extend the range of applications for which the general bag-of-frame approach gives satisfactory results, by proposing a mechanism that invents specific ad hoc features in an automatic way. We claim that these features can improve the performance of classifiers, independently of the details of the classification chain.

To find better features than the generic ones, one can find inspiration in the way human experts actually invent ad hoc features. The papers quoted above use a number of tricks and techniques to this aim, combined with intuitions and musical knowledge, in trial-and-error loops. For instance, one can use some front-end system to normalize a signal, or pass it through some filter, add pre or post-processing to isolate the (hopefully) most salient characteristics of the signal.

We propose here to automate a process of feature invention, with an algorithm which explores quickly a very large space of ad hoc functions. The functions are built by composing together - in the sense of functional composition - elementary operators. We call these functions analytical because they are described by an explicit composition of functions, by opposition to other forms of signal reduction, such as arbitrary computer programs.

This paper is structured as follows: In Section 2 we briefly introduce the EDS system, designed to create automatically and explore large sets of analytical features. Section 3 is devoted to the description of concrete experiments to assess the interestingness of analytical features, with regards to "generic" ones. These experiments can be seen as preliminary investigations of a feature space of billions (of billions) of features.

2 CREATION OF ANALYTICAL FEATURES: THE EDS SYSTEM

EDS – Extractor Discovery System – is a project – and a system – developed in the Sony CSL laboratory in Paris [39], [40], whose aim is to study experimentally the notion of analytical feature for audio signal processing applications.

EDS is able to explore efficiently the space of analytical features for arbitrary supervised audio classification problem. A problem is determined by a database of audio samples labeled (usually by hand) with a finite set of classes. The exploration of the space of analytical features is based on various function creation methods from a set of basic operators, considered as elementary. These two aspects are described in the following sections.

2.1 A library of elementary operators

The choice of elementary operators is of course arbitrary. These operators were selected so as to allow the creation of functions with a "reasonable" degree of abstraction, i.e. represent salient perceptive characteristics of the sounds with a relatively small number of operators (about 10, see below), while still allowing the possibility of exploring unknown, and possibly relevant functions. These operators are either basic mathematical operations (e.g. multiplication of a signal by a scalar, absolute value, max, etc.) or basic signal processing operators such as Fourier transforms, filters, *Db*, *root-mean-square*, and spectral operators like *spectralCentroid*, *spectralSkewness*, etc. This library also includes more specifically musical operators such as *Pitch* or *Ltas*. This list can of course be modified by the user. For the sake of reproducibility, we describe in this paper results obtained with the 76 basic operators listed in Annex 1.

The space of analytical features obtained by composition of basic operators is in principle infinite. One can however estimate the number of reachable functions by limiting the number of basic operators used in the expression of a function. With 5 operators maximum, the size of the feature space (with the basic operators of Annex 1) is approximately $2{,}5.10^{9}$. In practice, we explore functions with at most 10 operators, which represents a space of 5.10^{20} functions (hence the title of this paper). These sizes estimates were obtained through an exhaustive enumeration of all syntactically correct functions, and by discretizing scalar parameters (for instance we considered only 10 values for filter cut-off frequencies, etc.). We can observe that the number of functions grows exponentially with the number of operators used. Here are some typical examples of functions generated by EDS:

```
(A) Mean(Mfcc(Differentiation(x),5))
(B) Median(Rms(Split(Normalize(x),32)))
```

The first function (A) computes the average of the 5 first cepstral coefficients of the derivative of the signal (represented by *x*). The second one (B) computes the mean value (*Median*) of the energy (*Rms*) of successive frames (*split*) of 32 samples long in the normalized signal.
Feature creation is controlled by two mechanisms: typing and heuristics.

2.1.1 Typing

Each basic operator is typed according to the physical dimensions of its arguments. These types are used to avoid the construction of syntactically meaningless features. For instance, the Fft operator kaes as input something with a “time/amplitude” type, and its output type is frequency/amplitude. EDS can therefore generate *Fft(HpFilter(x))*, but not, e.g., *Fft(max(x))*.

2.1.2 Heuristics

These heuristics allow the system to further avoid exploring certain functions by evaluating them prior to the computation of their fitness. For instance, a heuristics states that it is usually unnecessary to consider functions with many times the same operator in a row, such as *fft(fft(fft(fft(x))))*.

In practice, adding a new basic operator to our library amounts to define 1) corresponding typing rules and 2) heuristics to control the use of this operator. There are other design issues related to adding operators, which are not covered in this paper (see rather Pachet & Zils Ismir04).

2.2 Creating analytical features

The creation of analytical features by composing elementary operators is based on genetic programming search [11]. The main steps of this search are the following:

1. Construction of an initial population of analytical features, by random compositions of operators.
2. Evaluation of features. This evaluation consists in computing the function on all the signals of the training database and compare these results with the labels. The correlation between these two series produces a *fitness* for the function. The computation of this correlation is described in Section 2.3.
3. Iteration of the process. The next population is built from the best features found in the current population, to which are added new features obtained using various genetic transforms of the current features.

The genetic transforms of step 3 are the following:

2.2.1 Substitution

Substitution consists in replacing one of the operators of the feature by another one, of compatible type. For instance:

(A’) *Max(Mfcc(Differentiation(x),5))*

is a substitution (*Max* replaces *Mean*) of (A)

2.2.2 Cloning

Cloning is a special case of substitution which consists in copying a feature but changing its numerical parameters, e.g.:

(B’) *Median(Rms(Split(Normalize(x),64)))*

is a clone of (B). Values of these parameters are chosen in a “reasonable” interval of values, which depends on the operator and the sampling rate.

2.2.3 Mutation

Mutation is an extension of substitution to sub expressions appearing in the definition of a feature, which satisfies the typing rules:

(A”) *Mean(Chroma(Normalize(x)))*

is a mutation de (A) in which the sub expression *Chroma (Normalize(x))* replaces *Mfcc (Differentiation (x),5)*.

2.2.4 Crossover

Crossover consists in combining two features to create a new one while satisfying the typing rules. For instance:

(C) *Mean(Rms(Split(Normalize(x),32)))*

(C’) *Median(Rms(Split(Differentiation(x)))*

(C) and (C’) are crossovers between (A) and (B).

2.2.5 Addition

Addition consists in adding an operator to the root of a feature:

(B”) *Abs(Median(Rms(Split(Normalize(x),32))))*

is an addition of (B).

2.3 Evaluation of individual features

To evaluate features, we need a computable criterion which measures the quality of a feature, i.e. its capacity to distinguish elements of different classes (in the case of a classification problem) or to approximate a target function (in the case of a regression problem).

In the case of classification, the *Fischer Discriminant Ratio* [5] is often used because it is simple to compute and reliable for binary problems (two classes). However it is notoriously inadapted to multi-class problems, in particular for non-convex distributions of data.

To improve the evaluation of features, we rather chose to implement an “embedded approach” (Blum and Langley, 1997) to feature selection, by which features are evaluated using a classifier built on-the-fly during the feature search process. The fitness of the feature is then defined as the performance of a classifier built with this unique feature (or more precisely its F-measure, [24]) trained on the training database. This measure yields better performance than the Fischer criteria on multi-class problems as the ones addressed in this paper.

In the case of a regression problem, sound samples are labeled with a real number rather than a class. These numbers are for instance the values of a target function one wants to approximate. The goal is to find features that allow to compute values that approximate this target function for new audio samples.

The most widely used criterion to assess a feature’s performance is Pearson’s Correlation, which we use for the regression problems presented here.

2.4 The space of analytical features

In this paper, we raise three questions about the nature of the analytical feature space:

1. Can analytical features capture information that generic features cannot?
2. Can all “perceptive” features be approximated by analytical features?
3. Can analytical features actually improve supervised classification tasks?

The sections below address these three questions through various experiments in the audio domain. To this aim, we consider a set of “generic features”, taken from the literature. This set is of course arbitrary, but is used here only as a means to compare analytical features empirically. The features of this reference set are listed in 8.2. In our experiments, we systematically compare the performance classifiers built with analytical features, with classifiers built with the generic feature set.

In order to assess the efficiency of analytical features, we compare them to results obtained with a “reference feature set”, whose complete list is given in Annex 3. This reference set includes general features commonly used in audio signal classification tasks, and well defined mathematically. The list includes notably the Mpeg-7 audio list, as well as several others, such as *Chroma*, often used for music analysis [7].

We systematically evaluate the performance of two classifiers: one built with the reference set, the other built with the analytical features found by EDS with the set of basic operators in 8.1. We follow the standard scheme in supervised classification: classifiers are systematically tested with samples that were not used either to create analytical features, or to train them. Moreover, we use 10-fold cross-validation wherever possible.

2.5 Choosing the classifiers

There is a huge literature concerning supervized learning algorithms [36], as well as many implementations of these algorithms, with no clear winner in the general case. In the aim of demonstrating the advantages of analytical features, we have conducted our experiments with various classifiers, to avoid biases (SVM, kNN, J48, neural networks). For the sake of clarity we report here only the results with Support Vector Machines (SVM) [29], which turned out to best and most stable of the algorithms we tried. We used SVM in the implementation of the Weka library (http://www.cs.waikato.ac.nz/ml/weka) with the default parameters, in particular the polynomial kernel.

We used EDS in a fully automated way for the creation and selection of analytical features. For each problem, we ran the genetic search until no improvements were found in feature fitness. The next section describes the feature selection process.

2.6 Feature Selection

To compare the two approaches (general versus analytical features) in a fair manner, it is important to train classifiers on spaces with identical dimension.

The set of all reference features (cf. Annex 2) has a dimension of 100. For each problem we explore a very large feature set (see below) from which we select feature sets of dimension less than or equal to 100. More precisely we present results obtained for various sizes of feature sets (from 1 to 100). As we will now see, EDS finds not only better analytical features but also feature sets of lesser dimension.

3 QUESTION #1: CAN ANALYTICAL FEATURES CAPTURE INFORMATION THAT GENERIC FEATURES CANNOT?

This question can be answered quite simply by considering a somewhat artificial problem, built to show the limitation of generic features.

The problem we consider (already briefly sketched in [19]) consists in detecting a sinus waveform in a given frequency range (say 0-1000Hz) mixed with a powerful colored noise in another frequency range (1000-2000Hz). Since the colored noise is the most predominant characteristic of the signal, we show that generic features are unable to detect the isolated sinus. For instance, when we look at the spectrum of a 650Hz sinus mixed with a 1000-2000Hz colored noise (Figure 1), the peak of the sinus is visible but not predominant, and is thus hard to extract automatically with general spectral features.

Of course, this problem is easy to solve by hand, when one knows how these signals are constructed: It suffices to apply a pre-filtering that cuts off most of the frequencies of the colored noise, so that the sinus emerges from the spectrum. As seen on Figure 2, the sinus peak emerges when the signal is low-pass filtered, and becomes easier to extract automatically.

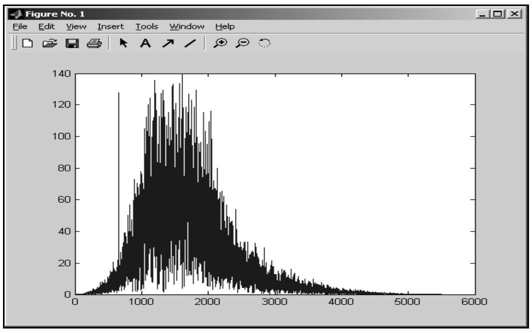

Figure 1. Spectrum of a 650Hz sinus mixed with 1000-2000Hz colored noise.

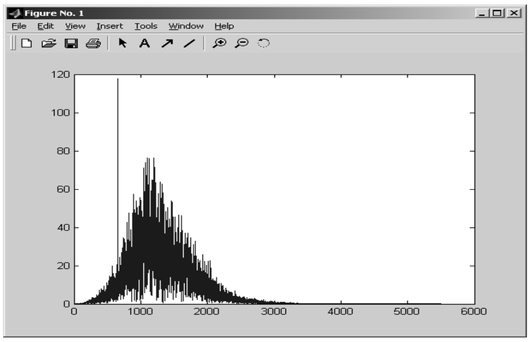

Figure 2. Spectrum of a 650Hz sinus mixed with 1000-2000Hz colored noise, pre-filtered by a 1000Hz Low-Pass Filter.

However, we show here that this particular example, although very simple, cannot be adequately solved using generic features. To this aim, we conduct the following experiment. We build a database of sinus + colored noise with varying values for the sinus (from 0 to 1000 Hz). Each signal is labelled with the corresponding frequency of the sinus.

We then train two "regression" classifiers for this task. The first one (REF) is trained using the generic feature set, with varying dimensions (from 1 to 100). The second one (EDS) is trained with analytical features built for this problem. EDS explored about 40,000 features to find an almost optimal analytical feature for this problem. The result is illustrated in Figure 3, and shows clearly that one analytical feature (of dimension 1) can solve the problem almost perfectly, whereas even the full set of generic features (of dimension 100) cannot. The best analytical feature found by EDS is the following:

(A) *MaxPos(Fft(Integration(BpFilter(Hamming(x), 10, 500))))*

This feature (A) is easy to interpret, and does almost exactly what should be done in this case, although probably not in the most simplest way. Its fitness (Pearson coefficient) is 9.99. The most likely human feature, knowing the database creation process, would be something like:

MaxPos(Fft(LpFilter(x, 1000)))

This "theoretically perfect" feature is also good (0.87), but slightly less than (A). This shows incidentally that the best features are not necessarily the most "justified".

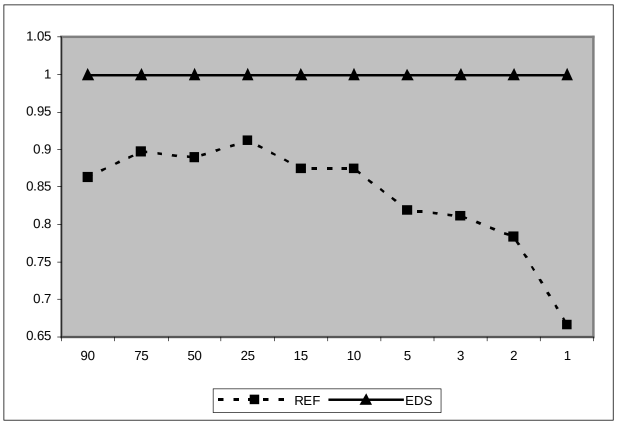

Figure 3. Results of the Pink noise experiment. One analytical feature is enough to solve the problem perfectly, whereas a feature set of dimension 100 with generic features cannot. The difference is even more drastic for smaller feature sets.

4 QUESTION #2: CAN ALL PERCEPTIVE FEATURES BE APPROXIMATED BY ANALYTICAL FEATURES?

This question is of course difficult to answer in a systematic way. We propose here also an experiment in which we compare the performance of generic features against analytical ones, but this time to approximate a “known” feature which is not in the reference set.

To this aim, we chose an audio feature called “spectral compactness”, an implementation of which is available from the jAudio feature extraction library.

This feature is described as follows (p. 603 of [15]): “Compactness is closely related to Spectral Smoothness as defined by [14]. The difference is that instead of summing over partials, compactness sums over frequency bins of an FFT. This provides an indication of the noisiness of the signal”.

We do not know the details of the implementation of compactness, nor its precise definition, but this is not needed here. The aim of this experiment is to see how “reachable” this feature is, from the set of generic features, and with analytical features.

To this aim, we also build several regression classifiers as in the preceding experiment. Because spectral features are known to be correlated, and also because MFCC are known to represent well spectral information in general, we conduct several experiments to show that spectral compactness is not easily reachable from other, known, sets of generic features. We used here a database of 2500 percussion sounds described in [25]. The results are illustrated in Figure 4 and Figure 5.

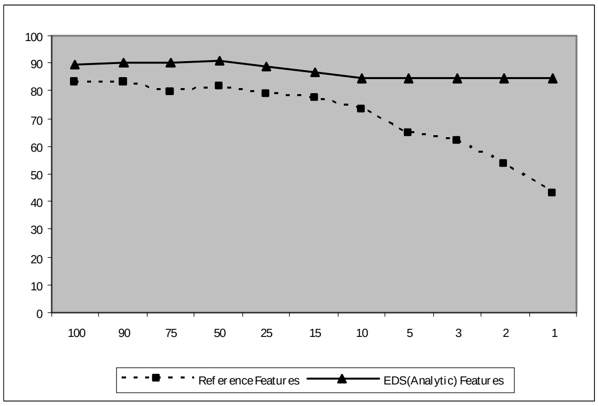

Figure 4. Results of the “Compactness” feature experiment. Compared performance between analytical and reference features, for various feature set dimensions. Analytical features perform better (and the improvement gets more important as the feature set dimension decreases).

Figure 4 shows again than the gain in performance of analytical features increases as the size of the feature set decreases, again with a spectacular result with single features (84% versus 43%).

More precisely, Figure 5 shows that analytical features also outperforms all the specific feature sets envisaged here: general spectral features, MFCC, and also the whole set of general features!

EDS produced the following analytical feature:

Abs (Sum (Integration (PeakPos (Fft (HpFilter (Derivation (x), 3969.0))))))

which yields a correlation coefficient of 0.79. This is an interesting result, since it contains operators that explicitly appear in the mathematical definition of spectral compactness (see definition above).

However, EDS produced an even better analytical feature: *Abs(Centroid(Hanning(Variance(Derivation(Split(x, 882)))))),* whose correlation with spectral compactness is over 0.84. We observe here the phenomenon described in Section 3.

The best reference feature found is: *HarmonicSpectralDeviation (Hanning (x))*

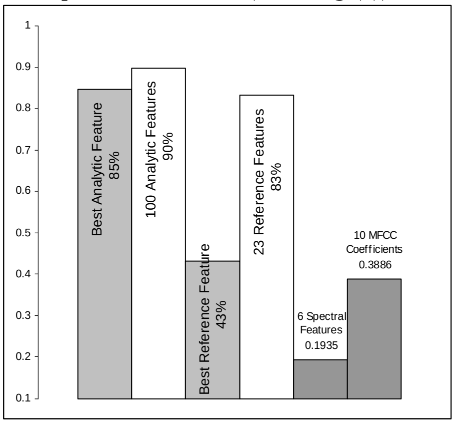

Figure 5. Comparison of various regression classifiers to approximate the “spectral compactness” feature. The best analytical feature yields a performance of 85%, better than the best reference feature (43%), far better than a feature set with 6 general spectral features (19%), and than the 10 first MFCC (38%), and slightly better than the whole set of reference features of dimension 100 (83%). A set of dimension 100 analytical features further improves the correlation up to 90%.

5 QUESTION #3: CAN ANALYTICAL FEATURES ACTUALLY IMPROVE SUPERVIZED CLASSIFICATION TASKS?

To address this general question we conducted several experiments on specific, multiclass audio classification tasks. These experiments are reported in various papers. For instance, we showed that analytical features can improve the classification of perceptive musical energy (Zils & Pachet 03), and, in a non musical context, dog barks (Molnar et al. 07). We report here the main conclusions of an experiment conducted on the classification of percussion sounds [25]. In this problem, we attempted to classify automatically sounds coming from a Pandeiro (Brazilian tambourine), using only very small portions of the attack of the sound. This problem arises from the desire to build new musical instruments that extend the possibilities given by traditional ones, using interactions with a computer. The results showed that here also, analytical features can improve the performance of classification over generic ones, as illustrated in Figure 6. Here again, the gain is particularly interesting for small size feature sets, as the classification performance is not substantially improved for high dimension feature sets. We think this glass ceiling is due to the inherent difficulty of the problem (attack portion of sounds may be inherently ambiguous in some cases).

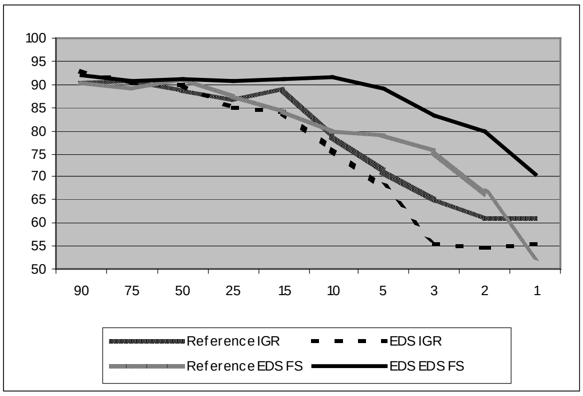

Figure 6. Classification results for the classification of attack portions of the Pandeiro into six sound classes. Compared performances of analytical features (EDS) with reference features (REF), using two feature selection methods.

6 CONCLUSION

We have introduced the notion of analytical feature for audio classification. Analytical features are created in a systematic and ad hoc way using the samples of the training database. Our feature generator is able to search in a space of about 10^{20} features. The main fundamental question we raised here is the nature of this space, and whether analytical features can cover characteristics of sounds that generic features cannot. To this aim we present two small experiments to compare the performance of classifiers using generic features and using analytical features. These experiments tend to show that analytical features do achieve, in general, better performance than generic features, or equivalent performance but with reduced feature sets. The experiments conducted here are not definitive as they use an arbitrary set of basic operators for producing analytical features, as well as an arbitrary set of reference features. However, we claim that analytical features should be investigated more systematically to increase our understanding of signal classification in general.

7 REFERENCES

[1] Aucouturier, J.-J, Defreville, B. and Pachet, F. The bag-of-frame approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. Journal of the Acoustical Society of America, 2007

[2] Boser, B. E., Guyon, I. M. and Vapnik, V. N. (1992) A training algorithm for optimal margin classifiers. In D. Haussler, editor, 5th Annual ACM Workshop on COLT, pages 144-152, Pittsburgh, PA. ACM Press.

[3] Chordia, P. Segmentation and Recognition of Tabla Strokes, ISMIR 20056, pp. 107-114, 2005

[4] Fiebrink, R. and Fujinaga, I. Feature Selection Pitfalls and Music Classification. ISMIR 2006, pp. 340-341, 2006

[5] Fisher, R. A. (1936) "The use of Multiple Measurements in Taxonometric Problems" Ann. Eugenics, vol. 7, pp. 179-186.

[6] Gillet, O. and Richard, G. (2003). Automatic Labelling of Tabla Signals. In Proceedings of the 4th International Symposium on Music Information Retrieval (ISMIR2003), Baltimore, Maryland.

[7] Gomez Gutierrez, E. (2006) Tonal Description of Music Audio Signals, PhD Thesis, Universitat Pompeu Fabra, Barcelone.

[8] Hanna, P., Louis, N., Dessainte-Catherine, M. and Benois-Pineau, J. Audio Features for Noisy Sound Segmentation. ISMIR 2004, 2004

[9] Herrera, P., Yeterian, A., Gouyon, F. Automatic classification of drum sounds: a comparison of feature selection methods and classification techniques. Proceedings of 2nd International Conference on Music and Artificial Intelligence, Edinburgh, Scotland, 2002

[10] Kim, H.G., Moreau, N, Sikora, T. (2005) Mpeg7 Audio and Beyond: Audio Content Indexing and Retrieval. Wiley & Sons.

[11] Koza, J. R. (1992) "Genetic Programming: on the programming of computers by means of natural selection", Cambridge, MA: The MIT Press.

[12] Lay Nwe, T. and Wang, Y. Automatic Detection Of Vocal Segments In Popular Songs. ISMIR 2004, 2004

[13] Liu, D., Lu, L. and Zhang, H.J., Automatic mood detection from acoustic music data, ISMIR 2003

[14] McAdams, S. (2002) Perspectives on the contribution of timbre to musical structure. Computer Music Journal, 23:85-102

[15] McEnnis, D. McKay, C., Fujinaga, I. Depalle, P. Jaudio: a feature extraction library, Ismir 2005.

[16] McKay, C. and Fujinaga, I. Automatic Genre Classification Using Large High-Level Musical Feature Sets. ISMIR 2004, 2004

[17] McKinney, M.F. and Breebart, J. Features for audio and music classification. ISMIR 2003

[18] Meng, A. and Shawe-Taylor, J. An Investigation of Feature Models for Music Genre Classification Using the Support Vector Classifier. ISMIR 2005: 604-609, 2005

[19] Pachet, F., Zils, A. Evolving Automatically High-Level Music Descriptors from Acoustic Signals. CMMR 2003: 42-53, 2003

[20] Pampalk, E., Flexer, A. & Widmer, G. (2005) Improvements of Audio-Based Music Similarity and Genre Classification (pp. 628-633), ISMIR 2005, ISMIR 2005, 6th International Conference on Music Information Retrieval London, UK.

[21] Peeters, G., Rodet, X. (2002) Automatically selecting signal descriptors for sound classification. Proceedings of the 2002 ICMC, Goteborg (Sweden).

[22] Peeters, G. (2003) Automatic Classification of Large Musical Instrument Databases Using Hierachical Classifiers with Inertia Ratio Maximization, 115th AES Convention New-York, NY, USA

[23] Peeters, G. (2004) A large set of audio features for sound description in the Cuidado project. http://recherche.ircam.fr/equipes/analyse-synthese/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf

[24] Rijsbergen, C. J. van (1979) Information Retrieval. Butterworths, London.

[25] Roy, P., Pachet, F. and Krakowski, S. Analytical Features for the Classification of Percussion Sounds: the Case of the Pandeiro, submitted to ISMIR 2007

[26] Sandvold, V. Gouyon and F. Herrera, P. (2004) Percussion classification in polyphonic audio recordings using localized sound models, ISMIR 2004.

[27] Scheirer, E. and Slaney, M. Construction and evaluation of a robust multifeature speech/music discriminator. Proc. ICASSP '97. 1997

[28] Schölkopf, B. and Smola, A. (2002) Learning with Kernels, MIT Press, Cambridge, MA.

[29] Shawe-Taylor, J. & Cristianini, N. (2000), Support Vector Machines and other kernel-based learning methods - Cambridge University Press.

[30] Sinyor, E., McKay, C., Fiebring, R., McEnnis, D. and Fujinaga, I. Beatbox Classification Using ACE. ISMIR 2005: 672-675, 2005

[31] Steelant, D. van, Tanghe, K., Degroeve, S., De Baets, B, Leman, M. and Martens, J.-P. (2004) "Classification of percussive sounds using Support Vector Machines", Proceedings of the annual machine learning conference of Belgium and The Netherlands, Brussels, Belgium.

[32] Tindale, A.R., Kapur, A., Tzanetakis, G. and Fujinaga, I. Retrieval of percussion gestures using timbre classification techniques, ISMIR 2004,2004

[33] Tzanetakis, G. Automatic Musical Genre Classification of Audio Signals, ISMIR 2001, 2001

[34] Tzanetakis, G. and Cook, P. (2002) Musical Genre Classification, IEEE Transactions on Speech and Audio Processing, Vol. 10, No. 5, July, pp. 293-301.

[35] West, K., Cox, S., 2004, "Features and Classifiers for the automatic classification of musical audio signals", Proc. ISMIR, pages 531-536, 2004

[36] Witten, I.H., Eibe, F. (2005) Data Mining: Practical Machine Learning Tools and Techniques. M. Kaufmann Publisher, 2nd Edition.

[37] Yoshii, K., Goto, M., and Okuno, H.G. AdaMast: A Drum Sound Recognizer based on Adaptation and Matching of Spectrogram Templates, ISMIR 2004, pp. 184-191, 2004

[38] Zils A., Pachet F., Delerue O., Gouyon F. Automatic Extraction of Drum Tracks from Polyphonic Music Signals. Proceedings of WEDELMUSIC2002, December 2002.

[39] Zils, A. & Pachet, F. Extracting Automatically the Perceived Intensity of Music Titles. Proceedings of the 6th COST-G6 Conference on Digital Audio Effects (DAFX03), September 2003. Queen Mary University, London.

[40] Zils, A. (2004) Extraction de descripteurs musicaux: une approche évolutionniste, Thèse de doctorat, Université Paris 6.

8 ANNEXES

8.1 Basic EDS operators

The list of basic operators used by EDS in this study is the following:

Abs	HFC	Pitch
Arcsin	HMean	PitchBands
AttackTime	HMedian	Power
Autocorrelation	HMax	Range
Bandwidth	HMin	RemoveSilentFrames
BarkBands	HpFilter	RHF
Bartlett	Integration	Rms
Blackman	Inverse	SpectralCentroid
BpFilter	Iqr	SpectralDecrease
Centroid	Length	SpectralFlatness
Chroma	Log10	SpectralKurtosis
Correlation	LpFilter	SpectralRolloff
dB	Max	SpectralSkewness
Differentiation	MaxPos	SpectralSpread
Division	Mean	Split
Envelope	Median	SplitOverlap
Fft	MelBands	Sqrt
FilterBank	Min	Square
Flatness	Mfcc0	Sum
Hamming	Mfcc	Triangle
Hann	Multiplication	Variance
Hanning	Normalize	Zcr
HarmSpectralCentroid	Nth	Harmonicity(Praat)
HarmSpectralDeviation	NthColumns	Ltas(Praat)
HarmSpectralSpread	PeakPos	
HarmSpectralVariation	Percentile	

A precise description of each operator can be found in (Zils, 04).

8.2 Annex 2 – Reference features

The list of general features used as the reference set is the following (features preceded by '*' could not be computed on the attack sounds because of their size):

```
* HarmonicSpectralCentroid(Hanning(x))
* HarmonicSpectralDeviation(Hanning(x))
* HarmonicSpectralSpread(Hanning(x))
Log10(AttackTime(x))
*Pitch(Hanning(x))
SpectralCentroid(Hanning(x))
* SpectralFlatness(Hanning(x))
SpectralSpread(Hanning(x))
Centroid(x)
PitchBands(Hanning(x),12.0)
Mfcc0(Hanning(x),20.0)
* HarmonicSpectralVariation(SplitOverlap(Hanning(x),2048,0.5))
Rms(x)
RHF(Hanning(x))
HFC(Hanning(x))
SpectralKurtosis(Hanning(x))
SpectralSkewness(Hanning(x))
SpectralRolloff(Hanning(x))
Iqr(x)
Chroma(Hanning(x))
MelBands(Hanning(x),10.0)
BarkBands(Hanning(x),24.0)
Zcr(x)
```