IMPROVING THE CLASSIFICATION OF PERCUSSIVE SOUNDS WITH ANALYTICAL FEATURES: A CASE STUDY

Pierre Roy
Sony CSL
6, rue Amyot
75 005 Paris
roy@csl.sony.fr

François Pachet
Sony CSL Paris
6, rue Amyot
75 005 Paris
pachet@csl.sony.fr

Sergio Krakowski[1]
Sony CSL Paris
6, rue Amyot
75 005 Paris
skrako@gmail.com

ABSTRACT

There is an increasing need for automatically classifying sounds for MIR and interactive music applications. In the context of supervised classification, we conducted experiments with so-called analytical features, an approach that improves the performance of the general bag-of-frame scheme without loosing its generality. These analytical features are better, in a sense we define precisely than standard, general features, or even than ad hoc features designed by hand for specific problems. Our method allows us to build a large number of these features, evaluate and select them automatically for arbitrary audio classification problems.

We present here a specific study concerning the analysis of Pandeiro (Brazilian tambourine) sounds. Two problems are considered: the classification of entire sounds, for MIR applications, and the classification of attack portions of the sound only, for interactive music applications. We evaluate precisely the gain obtained by analytical features on these two problems, in comparison with standard approaches.

1.Acoustic Features

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

The bag-of-frame approach [65535], [65535] 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.

MPEG7-audio ([65535]) as well as [65535], [65535] are standard sources for audio features. These features contain statistical information from the temporal domain (e.g. Zero-crossing rate), spectral domain (e.g. Spectral-Centroid), or more perceptive aspects (such as sharpness, relative loudness, etc.) and are mostly of low dimensionality.

The bag-of-frame approach has been used extensively in the MIR domain, for instance by [65535]. A large proportion of MIR related papers has been devoted to studying the details of this chain of process: feature identification [65535]; feature aggregation [65535]; feature selection [65535], [65535], [65535]; classifier comparison or tuning [65535], [65535]. This approach performs well on some problems, e.g. speech music discrimination. However, it shows limitations when applied to "difficult" problems such as genre classification, which works well on abstract, large categories (Jazz vs. Rock), but works poorly for more precise class problems (e.g. Be-bop vs. 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, [65535]), or by defining completely different schemes for classifying, e.g. the analysis-by-synthesis approach designed for drum sound classification [65535], and further developed by [65535] and [65535].

The bag-of-frame approach relies on generic features that do not always capture the relevant perceptive characteristics of the signals to be classified. Some classifier, like kernel methods [65535] including Support Vector Machines ([65535], [65535]) transform the feature space to increase inter-class separability. However, the increasing sophistication of feature selection algorithms or classifiers cannot compensate any initial loss of information.

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. For instance, one can use some front-end system to normalize a signal, or pass it through some filter, add pre-processing to isolate the most salient characteristics of the signal.

We have introduced in [65535] the EDS system, which automates feature invention. It used an evolutionary algorithm which explores quickly a very large space (about 10^{20}) of *ad hoc* features. The features are built by composing - in the sense of functional composition - elementary operators. We call these features analytical because they are described by an explicit composition of functions, as opposed to other forms of signal reduction, such as arbitrary computer programs. In the rest of this article when we refer to analytical features, we mean features invented by the EDS system.

[1] The work of Sergio Krakowski is partially supported by a CAPES scholarship.

2.PANDEIRO SOUND CLASSIFICATION

The Pandeiro is a Brazilian frame drum (a type of tambourine) used in particular in Brazilian popular music (samba, côco, capoeira, chôro). As it is the case for many popular music instruments, there is no official method for playing the Pandeiro. However, the third author, a professional Pandeiro player, has developed such a method, as well as a notation of the Pandeiro, that we use in this paper. This method is based on a classification of Pandeiro sounds in exactly six categories (see Figure 1): **tung** : Bass sound, also known as open sound; **ting**: Higher pitched bass sound, also open; **PA (or big pa)**: A slap sound, close to the Conga slap; **pa (or small pa)**: A medium sound produced by hitting the Pandeiro head in the center; **tchi** : The jingle sound; **tr**: A tremolo of jingle sounds.

The need for automatically analyzing Pandeiro sounds is twofold. First, MIR applications, for education notably, require the ability to automatically transcribe Pandeiro solos.

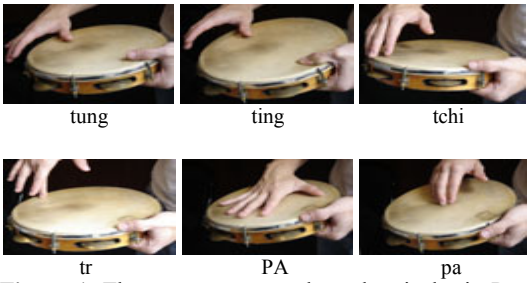

Figure 1. The gestures to produce the six basic Pandeiro sounds.

The second need is more original, and consists in developing real time interaction systems that expand the possibilities of the percussionist, to allow him to increase its musical "powers". In this case, we need to analyze robustly and quickly Pandeiro sounds, to trigger various events (see, e.g.[65535]).

We therefore define two different analysis problems, corresponding to these two applications.

The first problem consists in classifying complete sounds (150ms duration) in the six basic classes. The second problem, more difficult but more useful for real time applications, consists in classifying sounds using as less possible information, typically only the attack (about 3ms, that is 128 samples at 44 kHz), to allow a subsequent triggering of a musical event. To this aim we must build a reliable and very fast classifier.

2.1.Available sound databases

We have recorded 2448 complete Pandeiro sounds (408 of each 6 types) that constitute the *full sound* database. They were produced with the same instrument and recorded on a Shure Beta 98 microphone linked to a MOTU Traveller sound card.

In order to classify the sounds, it is important to finely locate them in time. To this aim, we designed a robust attack identifier, which works as follows: the incoming signal is divided in non-overlapping frames of 1.4ms (64 samples at 44kHz). A loudness value is computed for each frame, generating the "loudness curve". An attack is reported when a peak in this curve is found. The identifier is previously calibrated, in order to distinguish between noise peaks and real attacks.

For each attack, we record an audio file containing the attack frame itself and the following frame. This file populates the *attack* database.

2.2.Experiments: training and testing bases

We compare analytical features to a "reference feature set" [65535], containing standard acoustic features from e.g. Mpeg7-audio. We systematically evaluate the performance of two classifiers: one built with the reference set, the other built with EDS analytical features.

Each experiment is in turn divided in two parts. First, classifiers are trained on training samples and tested on the test samples. To this aim, databases are systematically divided in two parts, 2/3 for the training, and 1/3 for the test. The samples are chosen randomly, to avoid artifacts (e.g. evolution of the membrane during the recording session, small variations in the player gestures). In the second part, classifiers are trained and tested only on the test database, using 10-fold cross-validation.

This procedure aims at showing that the advantage obtained by analytical features is consistent, and do not depend on the conditions of experiments. The cross-validation using only the test database is motivated by the fact the EDS already uses the training database for evaluating the analytical features. So reusing it for training the classifiers could produce biases (although we are not sure why and how).

Finally, for the attack problem, we build an experiment in which the signal itself is used as a feature (this is possible because these signals are very short). The aim is to confirm that the signal is not a good feature.

2.3.Choosing the classifiers

There is a vast literature on supervised learning algorithms [65535] with no clear winner. We conducted experiments with various classifiers, to avoid biases (e.g. SVM, *k*NN, J48). For the sake of clarity, we report only the results with Support Vector Machines [65535], which turned out to be the best and most stable. (We use Weka's SMO with a 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. For the complete sound problem, EDS evaluated about 40,000 features. For the attack problem EDS evaluated about 200,000 features.

2.4.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. For the full sounds, all reference features could be computed, yielding a feature set of dimension 100. We have therefore selected 100 scalar analytical features among the 23,200 computed by EDS.

In the case of attacks, not all reference features were computable, because they are too small: only 17 reference features could be computed and evaluated, with a total dimension of the feature set of 90. We therefore selected 90 analytical features among the 53,500 EDS created for attacks.

We used two feature selection methods. Firstly, Information Gain Ratio (IGR) [65535], which corresponds to Weka's *AttributeSelection* algorithm with the following parameters: the *evaluator* is a *InfoGainAttributeEval* and the *search* is a *Ranker*, which allows us to determine a priori the dimension of the feature set. Secondly, we developed an algorithm suited to multi-class problems. The idea is to select a feature set that "covers" optimally the classes to learn from the viewpoint of individual features, that is, essentially of their F-measure. The algorithm iterates over all classes and selects features with the best F-measure for a given class.

Finally, we present results obtained for various sizes of feature sets from 1 to 100. This is an important aspect in the context of real-time systems, where we want to minimize the number of features to compute in real time. As we will see, EDS finds not only better features but also feature sets of lesser dimension.

2.5.Results and comments

Figure 2 and Figure 3 show the results obtained:

Experiment Description			Feature Set Dimension										
			100	90	75	50	25	15	10	5	3	2	1
Reference	*IGR*	*Train/Test*	99,9	99,9	99,6	99,5	99	99,5	99,1	92,8	88,5	65,2	56
Reference	*IGR*	*10-fold XV*	99,9	99,5	99,5	99,5	99,1	98,6	98,4	92	82	60,5	59,3
EDS	*IGR*	*Train/Test*	99,9	99,9	98,5	98,3	98,9	98,3	99,1	98	68,9	36,1	36,9
EDS	*IGR*	*10-fold XV*	99,9	99,9	99,9	98,8	98	98,4	98,2	97,8	64,7	36	21,2
Reference	*EDS FS*	*Train/Test*	99,9	99,9	99,9	99,8	99,1	99,1	98,9	98,8	93,6	80,8	67,2
Reference	*EDS FS*	*10-fold XV*	99,9	99,6	99,6	99,4	98,6	98,4	98,8	98,3	93,4	78,1	61,6
EDS	*EDS FS*	*Train/Test*	99,9	99,9	98,9	99,9	99,9	99,6	99,5	99	89,9	88,8	73,8
EDS	*EDS FS*	*10-fold XV*	99,9	99,9	98,9	99,7	99,6	99,5	99,4	99	91,3	89,5	73,6

Figure 2. Results on full sounds. **IGR** stands for Info. Gain Ratio. **EDS FS** denotes our F-measure-based FS algorithm. **Train/Test** and **10-fold XV** denote the experiments described in Section 2.2

Experiment Description			Feature Set Dimension								
			90	75	50	25	15	10	5	3	2	1
Reference	*IGR*	*Train/Test*	91,8	91,3	89,6	76,6	78,3	67,5	64,3	56,1	51,1	49
Reference	*IGR*	*10-fold XV*	92,6	91,2	88,8	79,9	73,2	67,4	64,7	44,2	42,4	34,5
EDS	*IGR*	*Train/Test*	95,1	93,3	92,3	77,7	72,5	63	61,3	54,7	54,5	56,9
EDS	*IGR*	*10-fold XV*	94,9	93,8	92,4	80,8	78,9	62,4	61	55,1	55,9	54,9
Reference	*EDS FS*	*Train/Test*	91,9	91,5	91	87,7	86,7	83,4	83,6	71,7	55,6	43,9
Reference	*EDS FS*	*10-fold XV*	91,9	91,5	90,2	86,1	85,2	78,9	82	68,5	48,6	39
EDS	*EDS FS*	*Train/Test*	94,9	94,4	94	92,1	91,4	87,9	90,1	88,6	80,4	72,1
EDS	*EDS FS*	*10-fold XV*	94,5	94	93,3	91,4	91,4	89	89,5	88	80,1	69,2
Signal			77.7	76.9	73.3	64.1	64.2	60	59.2	58.1	57.5	44

Figure 3. Results obtained with on attacks. See above for abbreviations. The "Signal" line gives the performance of classifiers using the input signal directly as a feature.

For the two problems, analytical features found by EDS improve the classification performance. The full sound problem is relatively easy. The use of the full reference feature set (dimension 100) yields a precision of about 99,9%. With the same dimension, analytical features yields the same precision. The gain becomes interesting if we consider feature sets of smaller dimension: 2 analytical features yield a precision of 89,5% versus 78% for general features.

Attack problems are more difficult and interesting. Analytical features are better than general ones, in particular for small feature sets. For attacks, 3 analytical features perform better than the 15 best general features.

Note that the gain depends on feature selection. IGR does not select the best EDS features for small feature sets (this is a known result [65535]). However, our feature selection algorithm yields better results for all sizes of the feature set, see Figure 4. This shows again the difficulty in interpreting directly the precision of classifiers.

The performance gain brought by analytical features for small feature sets has a lot of advantages, in particular for real-time applications. The 3 features that yield a precision greater than that obtained with 15 reference features are the following:

Abs (*Log* (*Percentile* (*Square* (*BpFilter* (x, 764, 3087)), 64)))

Centroid (*MelBands* (*Derivation* (*HpFilter* (*Power* (*Normalize* (x), 3), 100)), 6))

Abs (*Sum* (*Arcsin* (*Mfcc* (*Hann* (*HpFilter* (x, 19845)), 20))))

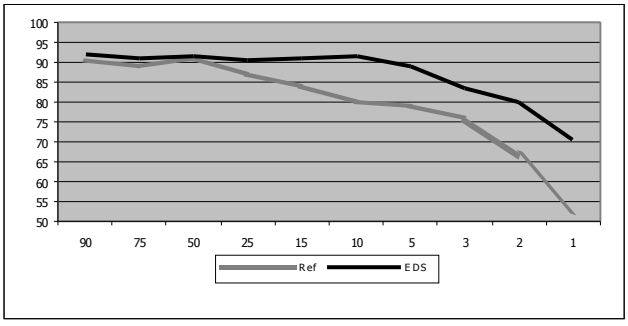

Figure 4. Analytical vs. reference features on attacks

This particular result allows us to consider real-time implementations: on a 3GHz Pentium IV PC, the computation of the 3 features for a 2,8 ms signal takes about

3 ms, to be compared to the computation of 15 generic features, which takes 9 ms, that is 3 times longer.

3.Conclusion

We have applied the EDS method for creating audio features, called analytical, to the classification of Pandeiro sounds. In both cases studied (full sounds, or only a portion of the attack) analytical features do improve the performance of classification, as compared to results obtained with generic, Mpeg-7 like features, in a bag-of-frame approach. The gain is notable both in terms of classification precision and feature set size. Moreover, the use of analytical features to improve classification algorithms may be combined with other optimization processes, such as boosting, bagging or *ad hoc* approaches.

5. REFERENCES

[65535] Aucouturier, J.-J. and Pachet, F. Tools and Ar*chitecture for the Evaluation of Similarity Measures : Case Study of Timbre Similarity*. ISMIR 2004.

[65535] 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. JASA, 2007.

[65535] Blum , A. and Langley, P. *Selection of Relevant Features and Examples in Machine Learning*, Artificial Intelligence, 1997 pp.245-271, Dec. 1997.

[65535] Boser, B. Guyon, I. and Vapnik V. *A training algorithm for optimal margin classifiers*. In D. Haussler, editor, 5th Annual ACM Workshop on COLT, pp.144-152, Pittsburgh, PA. ACM Press. 1992.

[65535] Fiebrink, R. and Fujinaga, I. *Feature Selection Pitfalls and Music Classification*. ISMIR 2006, pp. 340-341

[65535] Kim, H.G., Moreau, N. and T. Sikora Mpeg7 *Audio and Beyond: Audio Content Indexing and Retrieval*. Wiley & Sons. 2005.

[65535] Krarkowski, S. Pandeiro+, music video available at: http://www.skrako.com/eng/pop_video.html?aguas

[65535] McEnnis, D. McKay, C., Fujinaga, I. Depalle, P. *jAudio: a feature extraction library*, Ismir 2005.

[65535] McKinney, M.F. and Breebart, J. *Features for audio and music classification*. ISMIR 2003.

[65535] Peeters, G. and Rodet, X. *Automatically selecting signal descriptors for sound classification*. Proceedings of the 2002 ICMC, Goteborg (Sweden). 2002.

[65535] Peeters, G. *A large set of audio features for sound description in the Cuidado project*

[65535] Quinlan, J.R. *C4.5: Programs for machine learning*. Morgan Kaufmann. 1993

[65535] Sandvold, V. Gouyon, F. Herrera, P. Percussion classification in polyphonic audio recordings using localized sound models, ISMIR 2004

[65535] Scheirer, Eric D. and Slaney, Malcolm *Construction and evaluation of a robust multifeature speech/music discriminator*. Proc. ICASSP '97. 1997

[65535] Schölkopf, B. and Smola, A. *Learning with Kernels*, MIT Press, Cambridge, MA. 2002.

[65535] Shawe-Taylor, J. and Cristianini, N. Support Vector Machines and other kernel-based learning methods - Cambridge University Press. 2000.

[65535] West, K., Cox, S., *Features and Classifiers for the automatic classification of musical audio signals*, ISMIR 2004.

[65535] Witten, I.H. Eibe, F. *Data Mining: Practical Machine Learning Tools and Techniques*. M. Kaufmann Publisher, 2nd Edition. 2005

[65535] Yoshii, K., Goto, M., and Okuno, H.G. AdaMast: *A Drum Sound Recognizer based on Adaptation and Matching of Spectrogram Templates*, ISMIR 2004

[65535] Zils A., Pachet F., Delerue O., Gouyon F. *Automatic Extraction of Drum Tracks from Polyphonic Music Signals*. Proceedings of WEDELMUSIC, Dec. 2002

[65535] Pachet, F. and Roy, P. *Exploring billions of audio features.* In Eurasip, editor, Proc. of CBMI 07, 2007