# INCORPORATING PRIOR KNOWLEDGE ON THE DIGITAL MEDIA CREATION PROCESS INTO AUDIO CLASSIFIERS

*M. Lardeur, S. Essid, G. Richard*

TELECOM ParisTech
Institut TELECOM
37, rue Dareau - 75014 Paris, France

*M. Haller, T. Sikora*

Communication Systems Group
Technische Universität Berlin
EN 1, Einsteinufer 17, 10587 Berlin, Germany

## ABSTRACT

In the process of music content creation, a wide range of typical audio effects such as reverberation, equalization or dynamic compression are very commonly used. Despite the fact that such effects have a clear impact on the audio features, they are rarely taken into account when building an automatic audio classifier. In this paper, it is shown that the incorporation of prior knowledge of the digital media creation chain can clearly improve the robustness of the audio classifiers, which is demonstrated on a task of musical instrument recognition. The proposed system is based on a robust feature selection strategy, on a novel use of the virtual support vector machines technique and a specific equalization used to normalize the signals to be classified. The robustness of the proposed system is experimentally evidenced using a rather large and varied sound database.

***Index Terms—*** Audio processing systems, Learning systems, music processing

## 1. INTRODUCTION

Efficient audio classification systems should be able to exhibit some robustness to sound deformations due to varying content creation conditions. The latter include varying recording conditions, in particular heterogeneous room acoustics and sound capture techniques, and/or the application of common audio effects (such as expansion/compression, equalization, reverberation, etc.). Ideally, the classifiers are expected to be invariant under such deformations in the sense that their performance should not degrade when testing real world sounds which were recorded in different acoustic environments or processed by different audio effects, compared to the reference training sounds available at the lab.

Thus, the purpose of this work is to make classifying real world audio more efficient under highly varying audio creation processes, by incorporating some prior knowledge on these processes, especially the recording and post-production. To the best of our knowledge, our approach is completely novel. There have been a very few works on the robustness of some features used for musical signals classification to various deformations, especially "aggressive" ones which seriously alter the audio content, such as low bitrate mp3 coding, or noise addition [1, 2, 3]. However, there has been no previous concern with the influence of post-production audio effects on the classifiers behavior, nor any attempts to make use of this kind of prior knowledge to improve the classification. Note that this is significantly different from the classification of noisy signals, as widely studied in the speech/speaker recognition community [4].

We treat examplarily a realistic musical instrument classification scenario, where solo excerpts from real world commercial recordings are handled. It is important to note that our aim here is not to propose a high accuracy instrument recognition system, in marked contrast to other proposals [5, 6]. Rather, we focus on the approach to prior knowledge integration for robust audio classification, and merely apply it to the instrument classification problem

Figure 1 presents an overview of our approach. Our contributions are related to three classification stages, highlighted in light gray color. First, at the feature selection stage, a robust feature selection strategy, initially presented in a previous work [3] is re-used and further assessed. Second, at the classifier training stage, we introduce a novel use of the virtual Support Vector Machines approach [7]. Third, at the testing stage, a specific equalization is utilized to normalize the signals to be classified.

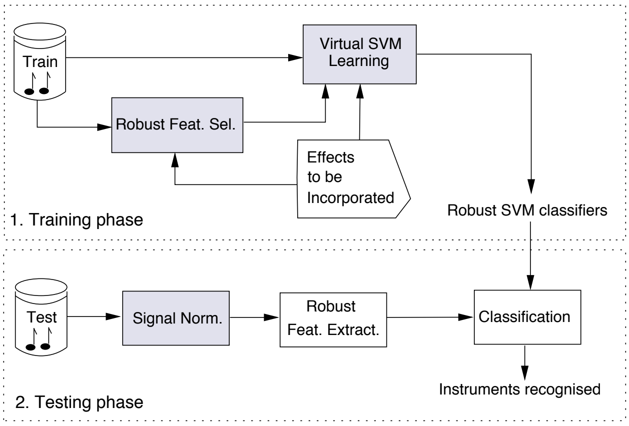

**Fig. 1**. System overview

This architecture results from extensive experimental work. In a preliminary phase (not described here) a wide range of audio deformations was considered and Support Vector Machine (SVM) classification experiments were undertaken on all sound versions (originals and deformed) which were used successively for training and testing. This allowed us to study how the application of the audio deformations influence both the learning process and decision making. In this paper, we show how we use the prior knowledge acquired during these preliminary experiments to implement a more robust classification system.

Following a brief description of the audio deformations considered and the feature extraction process in Sections 2.1 and 2.2, we recall the robust feature selection strategy in Section 2.3 before we

The research work has been supported by the European Commission under the IST FP6 research network of excellence K-SPACE.

introduce the use of the virtual support vector machines in Section 3. We proceed to the experimental validation in Section 4 and suggest some conclusions.

## 2. ACOUSTIC FRONT-END

### 2.1. Audio effects considered

Efforts have been dedicated to establish an inventory of audio effects commonly used in the audio creation process. It is worth noting that the application of some of these effects, typically reverberation, can be viewed as a way of simulating recording conditions whose parameters are in fact not directly available. With the help of an audio engineer, a subset of inescapable audio effects, both in the studio and live recording situations, have been chosen. The parameterization of the chosen effects have been made in such a way that the deformations remain perceptible and realistic, *i.e.* without drastically changing the timbre of the instrument sounds in a manner that would make them unrecognizable. The following three effects have thus been selected.
- *Reverberation* (or *reverb*), which can be seen as a way of simulating room acoustics, is probably one of the most utilized effects in post-production. It is applied especially when the microphone is placed close to the instrument or the voice that it is capturing. One way of applying reverb to sounds is by convolving the signals with a room impulse response. In our work, we have used the popular sox software [8], with the default reverb configuration.
- *Equalization* consists in attenuating or amplifying some spectral components of a sound, hence modifying its timbre. It is used by audio engineers for the correction of some recording defects (typically room and microphone defects, microphone misplacement, etc.), but also for aesthetic reasons as part of the artistic processes of audio mixing and mastering. We have implemented an equalizer in Matlab after [9] and configured it by mimicking the presets which are commonly suggested by popular audio players. Among the various configurations initially considered, we have retained a specific one that was used as a reference to generate others by multiplying the reference gains in each frequency channel by a constant. The gain curves of the equalizers which were obtained are depicted in Figure 2. The equalizer $EQ_2$ serves as a normalization bloc in the testing phase as shown in Figure 1. This will be further explained in Section 4.
- *Compression* is used to reduce the dynamic of recorded audio signals. Low energy signal portions are not modified while high energy ones are attenuated. Typically, it allows the audio engineer to accentuate the sustain parts of an instrument sound. We implemented our compressor after [10] and set the compression ratio to 10, the attack and release times respectively to 1ms and 1s, and the threshold to 0.5.

### 2.2. Feature extraction

We extract various audio features classicly used for our classification task [11, 5]. They include spectral, cepstral, temporal and perceptual features.
- The *spectral features* consist of the first four spectral statistical moments, the spectral irregularity and Octave Band Signal Intensities [5], the spectral slope, decrease, variation and frequency roll-off [11], as well as crest factors and MPEG-7 ASF (Audio Spectrum Flatness) [12].
- The *cepstral features* include Mel Frequency Cepstral Coefficients (MFCCs) and others computed with a Constant-Q Transform.
- The *temporal features* are the Zero Crossing Rates and temporal statistical moments as well as amplitude modulation features.

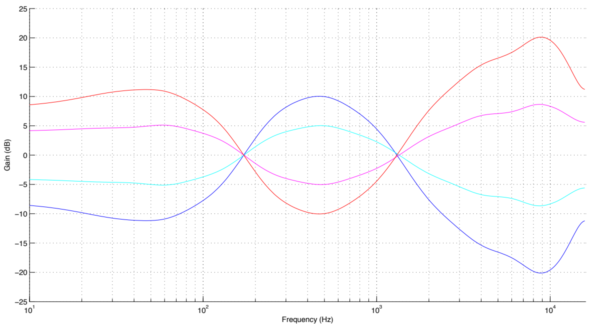

**Fig. 2**. Gain curves of the 4 equalizers used which will be referred to as $EQ_1$, $EQ_2$, $EQ_3$ and $EQ_4$ going from bottom to top in the low frequency region.

- Finally, the *perceptual features* consist of the relative specific loudness, sharpness and spread [11].
We thus get a total of 401 initial feature coefficients. All the features are extracted using 32-ms length frames with a hop size of 16 ms, except the AM features for which 960-ms length and 480-ms hopsize are used.

### 2.3. Robust Feature Selection (RFSA)

The baseline automatic Feature Selection Algorithm (FSA) which we use produces a ranking of the features based on a class separability criterion, *i.e.* a ratio of inter-class to intra-class separability measures (see [3] for the details). The $d = 30$ top-ranked features are then the ones which are selected. To make the FSA more robust, we prepare deformed versions of the training database, one version per considered deformation, and perform feature ranking over each one of these databases, in addition to the ranking of the features over the original training database. We thus obtain for each feature one rank per database instance and compute its robust rank as the average of these. This approach, which we originally presented in [3], has been further validated in this work (cf. Section 4) and will be referred to as RFSA.

## 3. VIRTUAL AUDIO SVM (VASVM)

SVM classifiers have proven efficient for a wide range of classification tasks and have become very popular in various research areas. We refer the reader to one of the many good tutorials on this powerful tool [13] and merely recall here the basic concepts which are referred to in the sequel. In bi-class problems, the SVM algorithm searches for the hyperplane $\mathbf{w}.\mathbf{x} + b = 0$ that separates the training samples $\mathbf{x}_1, ..., \mathbf{x}_n$ which are assigned labels $y_1, ..., y_n$ ($y_i \in \{-1, 1\}$) so that

$$y_i(\mathbf{x}_i.\mathbf{w} + b + \xi_i) - 1 \geq 0, \forall i, \tag{1}$$

under the constraint that the distance $\frac{2}{||\mathbf{w}||}$ between the hyperplane and the closest sample is maximal, $\xi_i$ being positive slack variables used to account for outliers. Vectors for which the equality in (1) holds are called support vectors. Since the data is not linearly separable in the original feature space, a kernel function $k(x, y)$ can be used to map the $d$-dimensional input feature space into a higher dimension space where the two classes become linearly separable. A test vector $\mathbf{x}$ is then classified with respect to the sign of the function $f(\mathbf{x}) = \sum_{i=1}^{n_s} \alpha_i y_i k(\mathbf{s}_i, \mathbf{x}) + b$, where $\mathbf{s}_i$ are the support vectors, $\alpha_i$ are Lagrange multipliers, and $n_s$ is the number of support vectors. Hence, the solution only depends on these support vectors.

Now our purpose is to incorporate prior knowledge about invariances in the classifiers, which can be achieved using the so-called *Virtual SVMs*. This technique was proposed by Decoste and Schölkopf and successfully applied to handwritten digits recognition [7]. The idea is to perform learning in three steps:
$1-$ classic SVM training is done on the set of training examples $\mathcal{X}$, yielding a set of support vectors $\mathcal{S} = \{\mathbf{s}_i\}_{1 \leq i \leq n_s}$;
$2-$ virtual training examples are generated by applying desired transformations to these support vectors, resulting in a set of new training examples $\hat{\mathcal{S}}$; the transformations are chosen so as to reflect some prior knowledge on the classification problem invariances;
$3-$ another training is performed on the set $\mathcal{S} \cup \hat{\mathcal{S}}$, yielding a new classifier that incorporates the invariances related to the transformations applied in step 2.

Following the same ideas, we proceed as follows:
$1-$ do SVM training on the original data $\mathcal{X}$;
$2-$ mark the audio frames corresponding to the support vectors found, apply the sound effects to them and extract the selected features from the transformed audio frames, thus creating the set of virtual training examples $\hat{\mathcal{S}}$;
$3-$ re-train the SVMs using the original feature vectors $\mathcal{X}$ plus all the virtual ones in $\hat{\mathcal{S}}$ created using all the effects.

We will refer to this approach as VASVM. Let us now present the experimental results which validate our classification strategy and discuss them.

## 4. EXPERIMENTAL VALIDATION

### 4.1. Experimental conditions

Six instruments are considered, namely, the Bassoon, Oboe, Violin, Cello, Guitar and Piano. Solo (unaccompanied) music was excerpted from commercial recordings of each instrument. There is a complete separation between sources from which the training excerpts were extracted and those providing the testing excerpts, a *source* being a music recording such that, either the recording studio, the artist or the instrument instance differs from one source to another. This allows us to assess the generalization capabilities of the classification system and observe how by incorporating invariances at the training stage, we are able to better classify the testing sounds which translate creation conditions that are significantly different from the ones related to the training sounds, thanks to this separation between sources. For each instrument class we use 22'54" of training data and 19'36" of testing data (the test database will be referred to as TDB). The number of training sources varies from 4 to 8 per instrument[1] and we use from 5 to 6 other testing sources per instrument. For the scoring, we use the average recognition accuracies over all the instruments. In fact, we classify 356-ms length segments (20 frames over which we perform early temporal integration whereby the mean of the corresponding 20 feature vectors is computed [6]) and compute the average accuracy over all these segments. Consequently, the resulting 95% confidence intervals for the accuracies that will be given are tight enough to allow us to consider 0.6% score differences as statistically significant. All the signals are downsampled at 32kHz and normalized to have zero mean and unit maximum absolute value. For our multi-class problem, a ”one vs one” strategy is used as in [5]. Additionally, we exploit the Gaussian kernel $k(x, y) = \exp\left(-\frac{||x-y||^2}{2d\sigma^2}\right)$. We tune the SVM hyper parameters, *i.e.* parameter $C$ and kernel parameter $\sigma$, by considering a grid of potentially useful values and performing 5-fold cross validation (where we use only the training data) to select the most appropriate ones. Note that optimal parameters are sought after for each of the following classification schemes that are tested.

In order to gain a deeper understanding of how the proposed enhancements act on the system performance, we have undertaken an intensive analysis of both the behavior of the features and the structure of the SVM classifiers, after the application of the effects. For the former, we have measured the feature statistics and performed visual analysis to compare 3D plots of original and transformed feature subsets (possibly after performing Principal Component Analysis for dimensionality reduction). For the latter, we have been concerned for instance with the “stability” of the support vectors, *i.e.* whether support vectors in the solution trained over the original training database remain support vectors when training is performed over transformed training sounds. These efforts have helped us answer much of our questioning as will be discussed hereafter.

### 4.2. Validation on the test set

All the effects described in Section 2.1 have been incorporated into the system, except the compression which was found not to degrade the performance of the reference system when applied to the test sounds. This finding is actually predictable since most of the audio features are hardly impacted by varying signal dynamics, in addition to the fact that the baseline system standardizes all the features [5] (zero mean and unit variance over the training database), hence this system is already invariant under dynamic compression.

Table 1 shows the improvement achieved by our enhanced classification system compared with the baseline system. Each one of our enhancements (highlighted blocs in Figure 1) brings a significant accuracy gain. We obtain a 5% accuracy improvement using the features selected by the robust FSA, the audio virtual SVM classifiers and the test sound pre-processing by the equalizer $EQ_2$. It is important to note that in our experiments the VASVMs have proven to be even more efficient than the SVM classifiers trained over the union of the original and all the transformed training databases (including all sounds). This can be explained by the fact that the SVM learning becomes more and more complex as the training database gets larger and larger. By augmenting the original database by only the virtual examples (which tend to stay around the original support vectors) the learning algorithm seems to converge to a more optimal solution. The other advantage of the approach is that one is not obliged to apply the effects to all the training sounds as it suffices to transform the original support vector frames. This may become critical if one wishes to incorporate more and more effects.

| System | Accuracy |
|---|---|
| Reference system | 75.3 |
| RFSA | 76.3 |
| RFSA+VASVM | 78.9 |
| RFSA+VASVM+$EQ_2$-Norm. | 80.8 |

**Table 1**. Average accuracy in % correct over the test database TDB. “$EQ_2$-Norm.” refers to the process of applying the equalizer $EQ_2$ to the test sounds.

[1] recall that collecting this type of data is very difficult hence one cannot always have as many sources as one might desire

### 4.3. Validation on an extended test set

To confirm that the system has incorporated the desired invariances, we applied the considered effects to the test database TDB (recall that it is distinct from the training database) and tested each one of the 5 new test databases (one per effect) with our improved classifiers (RFSA+VASVM). Table 2 sums up the results obtained.

Our strategy turns out to be effectively robust to audio effects. While the performance of the reference system may seriously degrade on some of the transformed databases, especially TDB+$EQ_4$, the accuracy of our improved system remains always greater than the reference system accuracy on TDB. Moreover, the mean accuracy of our proposal over all the test databases is more than 3% greater than the reference. Again both the robust FSA and virtual SVM training appear to be advantageous, although a stronger contribution to the improvement is brought by the VASVM approach.

| Test data | Reference | RFSA | RFSA+VASVM |
|---|---|---|---|
| TDB | 75.3 | 76.3 | 78.9 |
| TDB + reverb | 73.3 | 73.6 | 76.7 |
| TDB + $EQ_4$ | 70.6 | 71.2 | 75.9 |
| TDB + $EQ_3$ | 73.8 | 75.0 | 78.3 |
| TDB + $EQ_2$ | 77.7 | 78.4 | 80.8 |
| TDB + $EQ_1$ | 78.1 | 78.7 | 80.6 |
| Mean | 74.8 | 75.5 | 78.5 |

**Table 2**. Average accuracy in % correct over the TDB and its 5 versions transformed with the audio effects considered, using the same improved classifiers RFSA+VASVM.

Also of note is the fact that the scores over the test sounds transformed by $EQ_1$ and $EQ_2$ are always greater than the ones measured over the remaining databases. This observation has motivated us to systematically pre-process the test signals by $EQ_2$ in our final system. It is worth mentioning that the benefits of this pre-processing stage have been further validated on a third completely different database (distinct from the training and the testing databases presented here) as we observed again a greater performance after $EQ_2$-normalization, compared to no pre-processing of the sounds to be classified. From the gain curves presented in Figure 2 it can be seen that the equalizers $EQ_1$ and $EQ_2$ tend to emphasize the spectral components between 150 and 1500 Hz where most of the first partials of music notes occur, on average over many excerpts. It is difficult to interpret why this would be beneficial to the classification performance. Yet, through our analysis we have been able to work out that the application of $EQ_2$ and $EQ_1$ tend to decrease the features intra-class variance, in contrast to other effects, hence making the features more stable.

Another interesting question is: how should one choose the effects to be incorporated? We suggest applying the candidate ones to the testing database to check whether a reference classifier performance degrades on the transformed sounds. If no accuracy degradation is observed (as it was the case with the compression in our study) there is obviously no need to incorporate those effects. The more important question is how to choose the selected effects parameters? We have been able to verify that it is important to choose complementary parameters as we did with $EQ_3$ and $EQ_4$ which are the symmetric versions of $EQ_1$ and $EQ_2$. The average accuracy on all the test databases falls to 76.3% when $EQ_1$ and $EQ_2$ are incorporated without $EQ_3$ and $EQ_4$, and to 74.4% when proceeding the other way round.

## 5. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a method for incorporating prior knowledge on the process of music recording and post-production into audio classifiers. By choosing relevant audio effects, selecting robust features, performing virtual audio SVM training and normalizing the sounds to be classified using a specific equalizer, one can achieve significantly better classification performance, compared with a standard approach. The improved system becomes invariant under the effects incorporated, hence more robust under varying media creation conditions. Up to 5% improvement in the recognition accuracy of an instrument classification system was obtained with the proposed method.

Future work will look at incorporating more audio effects, and chiefly the optimal way of superposing the various effects. We will also try to validate the method on other audio classification problems.

## 6. REFERENCES

[1] S. Sigurdson, K.B. Petersen, and T. Lehn-Schioler, “Mel frequency cepstral coefficients: An evaluation of robustness of mp3 encoded music,” in *ISMIR*, Vicoria,Canada, Oct. 2006, pp. 286–289.

[2] D. Stowell and M. D. Plumbley, “Robustness and independence of voice timbre features under live performance acoustic degradations,” in *11th Int. Conference on Digital Audio Effects*, Espoo, Finland, Sept. 2008, pp. 325–332.

[3] S. Wegener, M. Haller, J.J. Burred, T. Sikora, S. Essid, and G. Richard, “On the robustness of audio features for musical instrument classification,” in *EUSIPCO*, Lausanne, Switzerland, Aug. 2008.

[4] G. M. Davis, Ed., *Noise Reduction in Speech Applications*, CRC Press, 2002.

[5] Slim Essid, Gaël Richard, and Bertrand David, “Instrument recognition in polyphonic music based on automatic taxonomies,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 68–80, Jan. 2006.

[6] G. Richard C. Joder, S. Essid, “Temporal integration for audio classification with application to musical instrument classification,” *To appear in IEEE Transactions on Audio, Speech, and Language Processing*, 2009.

[7] Dennis Decoste and Bernhard Schölkopf, “Training invariant support vector machines,” *Mach. Learn.*, vol. 46, no. 1-3, pp. 161–190, 2002.

[8] SoX Sound eXchange, “sox-14.0,” http://sox.sourceforge.net/.

[9] Udo Zölzer, “Audio processing systems,” Tech. Rep., Technical University of Hamburg, Jan. 1997.

[10] Udo Zölzer, *DAFX-Digital Audio Effects*, John Wiley & Sons, 2002.

[11] Geoffroy Peeters, “A large set of audio features for sound description (similarity and classification) in the cuidado project,” Tech. Rep., IRCAM, 2004.

[12] ISO/IEC, “Information technology - multimedia content description interface - part 4: Audio,” International Standard ISO/IEC FDIS 15938-4:2001(E), ISO/IEC, June 2001.

[13] B. Shölkopf and A. J. Smola, *Learning with kernels*, The MIT Press, Cambridge, MA, 2002.