MULTI-SCALE TEMPORAL FUSION BY BOOSTING FOR MUSIC CLASSIFICATION

Rémi Foucard(1)**, Slim Essid**(1)**, Mathieu Lagrange**(2)**, Gaël Richard**(1)

(1)TELECOM ParisTech, CNRS-LTCI
37, rue Dareau
75014 Paris, France
remi.foucard@telecom-paristech.fr

(2)Ircam, CNRS-STMS
1, place Igor Stravinsky
75004 Paris, France

ABSTRACT

Short-term and long-term descriptors constitute complementary pieces of information in the analysis of audio signals. However, because they are extracted over different time horizons, it is difficult to exploit them concurrently in a fully effective manner. In this paper we propose a novel temporal fusion method that leverages the effectiveness of a given set of features by efficiently combining multi-scale versions of them. This fusion is achieved using a boosting technique exploiting trees as weak classifiers, which has the advantage of performing an embedded feature selection. We apply our algorithm to two standard classification tasks, namely musical instrument recognition and multi-tag classification. Our experiments indicate that the multi-scale approach is able to select different features at different scales and significantly outperforms the mono-scale systems in terms of classification performance.

1. INTRODUCTION

Automatic classification of audio signals is one of the main research areas in the field of music information retrieval. This task consists in assigning audio signals to one or more categories (classes), according to a chosen criterion, which can be the musical instrument played, the speaker gender, the corresponding musical genre, etc. Classification can be very useful for many applications scenarios, such as database annotation, stream segmentation, and smart organization and search of large libraries.

Most audio classification systems represent the signal by splitting it into fixed-duration frames, from which several features are computed to be used by a learner. Given such training examples, the learner will then build a rule for determining the relevant class of any previously unseen example, only by considering its features. However, using frames of the same length limits the duration of the observable phenomena. While describing signal characteristics at different scales has become frequent in image processing [15], few audio-related studies use several temporal horizons for describing the signal.

The purpose of the present work is to setup a classification scheme that leverages the discrimination power of the features considered, by extracting them at different time scales and using a boosting technique to combine them efficiently. To precisely demonstrate the advantage brought by the use of different scales, we keep the same representation at every scale (*i.e.* compute the same features at different scales), but our system is flexible enough to handle different types of descriptions through varying scales.

In the remainder of this paper, we first briefly review audio classification algorithms and related temporal integration techniques in Section 2. Then we describe our multi-scale classification method (Section 3), and in Section 4, we present our experiments and results.

2. RELATED WORK

Audio classification makes use of machine learning to build rules for predicting the relevant class of a previously unknown audio excerpt. A good overview of the music classification problems and most common techniques can be found in [5].

First, the signal is described by a set of features. Among the most common, we can name: Fourier transform coefficients, mel-frequency cepstral coefficients (MFCC), delta-MFCC, chromagrams or zero-crossing rates [17]. Most of the time, several features are computed from a single frame, then they are concatenated into one high-dimensional feature vector.

In order to map the obtained description to class labels,

This work was realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation

various classifiers have been considered in previous works. The two most used ones are probably *Gaussian mixture models* (GMM) [16] and *Support vector machines* (SVM) [11]. Alternatively, several recent works have made use of boosting, a meta-classifier training several complementary versions of other learners [3, 4].

Most systems choose to represent the signal using fixed-length frames. However, the concepts behind each class may be conveyed by signal properties that have heterogeneous temporal dynamics. Therefore, potentially useful descriptors may need to be built at various time scales. Hence, a problem occurs when one tries to fuse such descriptions, because simple concatenation of the features (as done in most works) is infeasible.

Early integration [9] can be used to solve this problem, simply by integrating the features computed over shorter frames, over the duration of the longest analysis window. This synchronization of all descriptors allows for their concatenation, but the temporal precision of the shorter-term features is reduced. Therefore, potentially useful high-frequency content is lost, due to the integration low-pass filtering effect.

In [2], the authors fuse MFCC, along with chroma, web documents analysis and Last.fm tags [1], by means of kernel fusion. The boosting algorithm can also be used for classifier fusion [18]. In [1], fusion by boosting is applied to audio data, but all representations are done at the same scale: one vector per song. We can also cite [12], where the authors discriminate speech/nonspeech segments with features built using a constant-Q filterbank. In this kind of transform, the filters do not usually have the same temporal support. However, once the feature vector is built, no information is kept about the temporal support.

Furthermore, studies pointed out that representing the signal on different scales, and jointly considering all scales during the whole learning process, may lead to a more complete analysis of the signal than using a single temporal horizon [14]. Indeed, short-term features can precisely capture short events and quick changes in the signal. On the other hand, long-term features are able to represent larger phenomena, but with a poor temporal resolution. Using features built over several scales should then allow for describing jointly more diverse aspects of the signal.

3. PROPOSED METHOD

We propose a novel boosting scheme to achieve multi-scale information fusion at a decision level. The boosting algorithm trains a *weak* classifier several times, putting the emphasis on different examples among iterations. As mentioned in Section 2, boosting has already been adapted for classifier fusion. This fusion can be achieved by simply considering several weak classifiers in parallel, and selecting, at each iteration, the best performing one. This constitutes a convenient framework for heterogeneous classifier fusion because it does not make any assumption on the nature of the weak classifiers. It considers only their decisions on the training examples.

3.1 Multi-scale representation

In this work, we evaluate the merit of a multi-scale feature representation compared to the classical mono-scale representation. In order to clearly identify the usefulness of the multi-scale approach compared to the mono-scale one, the same set of features is used at every scale. Further details on the features used are given in Section 4.

The multi-scale feature representation is built as follows. First, the sequence of descriptors is computed at the finer scale, and then the other ones are obtained by temporal integration, which allows for fast feature computation. We integrate feature vectors by temporal averaging.

3.2 Boosting trees

For every scale s, our weak learner $\mathcal{H}_s$ is a CART [2] classification tree [7] using L_s-sample length frames. Trees are convenient, as they can be trained fast, and have proven efficient when boosted [3]. Furthermore, they present the advantage of performing feature selection during their training.

Decision trees are built from a root containing all training examples. At each node, the data is split in two (or possibly more) children nodes, only using a threshold on a particular bin of the feature vector. The bin and threshold values are chosen so that the two children nodes are the “purest” possible (*i.e.* the probabilities of the two classes are the furthest possible from 0.5).

Here, we use binary trees, with the Gini impurity measure, which is the most common configuration. The depth is fixed in advance, and we separately experiment depths 1 (which is also referred to as a stump) and 2.

3.3 Decision ranges

At each boosting iteration, the boosting algorithm chooses the weak classifier with the lowest weighted error rate. Making a fair comparison between the classifiers implies that the decisions, for each of them, must be taken on the same audio segments. Because the frames of the different classifiers do not describe the same portions of signal, we have to set the length on which the decisions are taken, for all scales.

[1] Last.fm is an online music listening service, where any user can associate any tag to a song. These tags can be automatically retrieved through an API.

[2] Classification And Regression Tree.

For this purpose we introduce *decision ranges*. These ranges represent the portions of signal on which the decisions of the weak classifiers are taken. Figure 1 shows how a decision range i, in gray, includes the feature frames from the different scales. Each $\mathbf{x}^n_{i,s}$ is a description vector, where s is the temporal scale level, n is the index of the frame within the decision range, and i represents the surrounding decision frame. We consider a frame to belong to range i if its center is included in the temporal bounds of i. In the following, we will denote by $\mathbf{x}_i$ the set of all frames from all scales, that belong to range i.

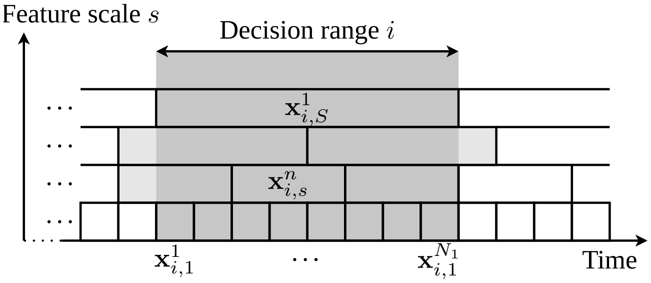

Figure 1. A decision range (in gray), covering a different number of frames on different scales.

Figure 1 also shows that a decision range cannot be shorter than the frames at the largest scale. Otherwise, the largest scale could be favored because it uses a greater amount of signal. On the contrary, decision ranges longer than $L_{\max}$ would decrease the number of training examples. This is why each range spans exactly $L_{\max}$ samples.

3.4 Core algorithm

The whole learning procedure is detailed in Algorithm 1.

We start from the examples $\mathbf{x}^n_{i,s}$, with class labels y_i. The labels neither depend on s nor n, but only on the current song which comprises segment i, as we are assuming class labels always span the whole song duration. Thus, although final decisions may be taken at a song-level, they are obtained by combining intermediate decisions taken on segments of a song, referred to as *decision ranges*, based on a corresponding set of feature-vector instances $x^n_{i,s}$.

Each of these decision ranges gets an associated weight, representing the relative focus of the algorithm during the current iteration. In the beginning, all weights are equal for ranges belonging to the same class.

At each iteration r, the weights $w_{r,i}$ are normalized so they sum to 1, before the weak classifiers $h_{r,s}$ (the CART trees) are trained. These trainings must take into account the weights of the examples. For each scale, the decision on range i is a majority vote on all frames belonging to i. Using these decisions, we can compute an error rate for every

Algorithm 1 Adaboost for multi-scale classifier fusion.

```
Input: Annotated examples from all scales $(\mathbf{x}^n_{i,s}, y_i)$,
    $1 \leq i \leq I, \quad 1 \leq s \leq S, \quad 1 \leq n \leq N_s$
Input: Weak learners $\mathcal{H}_s$
  $w_{1,i} \longleftarrow \frac{1}{2m}, \frac{1}{2l}$, resp. for $y_i = 0, 1$, where $m$ and $l$ are the number of negative and positive examples, respectively

  for $r = 1, \ldots, R$ do
    $w_{r,i} \longleftarrow \frac{w_{r,i}}{\sum_{j=1}^{I} w_{r,j}}$                // Normalize the weights
    Train classifiers $h_{r,s}$ with the models $\mathcal{H}_s$ and weights $w_{r,i}$

    // Decisions of $h_{r,s}$ on the observation ranges $i$
    $d_{r,s,i} = \begin{cases} 1 & \text{if} \quad \frac{1}{N_s} \sum_{n=1}^{N_s} h_{r,s}(\mathbf{x}^n_{i,s}) > 0.5 \\ 0 & \text{otherwise} \end{cases}$ ,

    // Compute weighted error rate
    $\epsilon_{r,s} \longleftarrow \sum_i w_{r,i} \, |d_{r,s,i} - y_i|$

    // Best scale
    $\hat{s}_r \longleftarrow \operatorname{argmin}_s \epsilon_{r,s}$
    $\epsilon_r \longleftarrow \epsilon_{r,\hat{s}_r}$
    $h_r \longleftarrow \sum_n h_{r,\hat{s}_r}$
    // Coefficient associated with $h_r$
    $\alpha_r \longleftarrow \log \frac{1}{\beta_r}$, where $\beta_r = \frac{\epsilon_r}{1-\epsilon_{r,}}$

    // Update the example weights
    for all ranges $i$ do
      // test whether  $d_{r,\hat{s}_r,i} = y_i$
      if $\mathbf{x}_i$ well classified then
        $w_{r+1,i} \longleftarrow w_{r,i} \, \beta_r$
      else
        $w_{r+1,i} \longleftarrow w_{r,i}$
      end if
    end for
  end for

Output: $H(\mathbf{x}) = \sum_r \alpha_r h_r(\mathbf{x})$
```

scale. The scale $\hat{s}_r$ with the lowest error rate is then selected for the final strong decision, with weight α_r. After that, the weights of the correctly classified examples are decreased, thus reducing their importance for future iterations.

The final output $H(\mathbf{x})$ is used during the testing phase as follows. When tagging a range i, one decision is taken for each component r by applying h_r to the observations from corresponding scale $(\mathbf{x}^n_{i,\hat{s}_r})$. Then, $H(\mathbf{x}_i)$ is a weighted sum of the $h_r(\mathbf{x}_i)$, as stated at the end of Algorithm 1. Finally, the global decision for a whole song a is a standard late integration over all decision ranges within a. It is done by taking the thresholded mean of the $H(\mathbf{x}_i)$:

$$D_a = \begin{cases} 1 & \text{if} \quad \mathrm{mean}_{i \in a} H(\mathbf{x}_i) > t \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

4. EXPERIMENTS

To show the usefulness of our multi-scale system compared to mono-scale systems, we perform experiments on two datasets, corresponding to two usual tasks is audio classification. We first validate our method on a musical instrument recognition database. Then, we test our system performance for multi-tag classification on the now well-known CAL500 [16]. The two experiments are done with different sets of features and different scale choices.

4.1 Musical instrument recognition

The task of instrument recognition presents the advantages of being well defined and strongly related to the audio content. This is why we run the first experiment on a database containing a set of solo real-music performances, featuring six instruments: Piano, Guitar, Bassoon, Oboe, Cello and Violin. The database contains 73 files (31 for training, 42 for testing), totalling 449 minutes of music. For each instrument, we have between 28 and 39 minutes of performance in the training set, and between 22 and 64 minutes in the test set.

From this data, we extract a selection of 30 feature coefficients obtained by applying Inertia Ratio Maximisation [13] to an initial set of cepstral, spectral, perceptual and temporal features used in a previous work [10].

We extract these descriptors at four distinct scales. The shortest one ($S1$) has an analysis window of $L_1 = 320$ ms, which is approximately the duration of an eighth note at 90 BPM. The other scales ($S2$, $S3$ and $S4$) have windows of lengths $2L_1$, $4L_1$ and $8L_1$. The frames do not overlap.

On this data, we trained our systems with 500 boosting iterations, using trees of depth 1.

Each example is annotated with one of the six instruments. We decompose this multiclass problem into six distinct bi-class problems, following the one-versus-all approach. During the test phase, all decisions are integrated to the largest scale $8L_1 = 2.6$ s, and the most probable instrument is chosen. For the mono-scale systems with scales shorter than $8L_1$, the late integration is done by summing the classifier output on the frames within the considered decision range.

With these predictions on the test set, we calculate the recognition rate as:

$$R = \mathrm{mean}_i \ \mathbb{1}_{H(\mathbf{x}_i)=y_i} \tag{2}$$

4.2 Multi-tag classification using CAL500

For this experiment, we use the CAL500 database [16], a database containing 500 pop songs, annotated by non-experts through a survey. Each song has been annotated by at least three people. We keep the 61 tags used in [2]. These tags describe different properties of the whole songs, such as: mood, genre, instrument, *etc*.

Tests are conducted using 10-fold cross-validation, with 450 songs used for training, and 50 songs for testing. The test sets are not overlapping between the different folds. For complexity reduction, we only use 30s of each song: extracted between instants 30 s and 60 s.

The features we use for describing each frame of signal are: the 15 psychoacoustic-related features recommended in [19] (loudness, tonal dissonance, ...), completed by the common first 13 MFCC (dropping the energy), chroma, zero-crossing rate, and spectral spread, skewness and kurtosis.

We have chosen five different scales: frames covering 2, 3.3, 5.5, 9 and 15 s of signal, with 50% overlap. A preliminary experiment indicated that, for this kind of data, scales under 2 s were less useful. And we also considered that 15 s was long enough to capture a wide range of long-term phenomena. The other scales are chosen to have a constant logarithmic spacing between each consecutive values.

We examine the performance on the test set, with 100 boosting iterations, using the same two evaluation measures as in [2]. These ranking metrics measure the ability of a soft prediction system to output higher scores for relevant documents compared to irrelevant ones. Soft predictions are non-binary scores, representing the amount of confidence the predictor has in the positive association of a considered tag to a given song. We can obtain soft outputs from our system, simply by averaging instead of thresholding the final decision:

$$\tilde{D}_a = \mathrm{mean}_{i \in a} H(\mathbf{x}_i) \tag{3}$$

This framework will make performance evaluation independent from the detection threshold t, that we choose for Equation 1.

From these decisions, we compute the Mean Average Precision (MAP) and Area under the ROC[3] curve (AUC). For a precise description of their calculation, see [8].

[3] Receiver Operating Characteristic.

Scale	Recognition rate (in %)
$S1$	59.8
$S2$	53.0
$S3$	62.9
$S4$	44.2
Multi-Scale	64.5

Table 1. Performance of the different systems on the instrument recognition database.

Scale	Tree depth	MAP	AUC
Scale 1	1	0.432	0.641
	2	0.449	0.653
Scale 2	1	0.442	0.652
	2	0.454	0.660
Scale 3	1	0.448	0.658
	2	0.451	0.662
Scale 4	1	0.456	**0.667**
	2	**0.458**	0.667
Scale 5	1	0.457	0.664
	2	0.451	0.661
Multi-Scale	1	**0.466**	**0.671**
	2	0.458	0.665

Table 2. Performance of the different systems on CAL500.

4.3 Results and discussion

The recognition rates yielded by the different systems on the instrument database are presented in Table 1. It is found that the multi-scale system has the best recognition rate. The difference between multi-scale and scale 3 systems is significant, according to a McNemar test [6], which yielded a p-value of 0.003. This means that the difference is statistically significant with a 99.7% confidence level.

The features selected by the trees along the boosting iterations differ greatly from one instrument to another, but the most selected scales are the shortest and the longest ones ($S1$ and $S4$). Surprisingly, these two scales do not correspond to the best performing mono-scale systems. This may be due to the fact that $S1$ gives the most temporally precise description, while $S4$ is good at taking decisions on a 2.6 s decision range, since it has the same length. Most of all, this indicates that the information brought by the whole set of scales is structurally different from just one scale.

A closer look at the detailed results, on a per-instrument basis, also revealed that the multiscale system is not the best performing one for all instruments. However, its performance is less variable among instruments. This shows that the multi-scale approach performs best, as it is more flexible, and can focus on the most appropriate representation.

The results for the multi-tag task on CAL500 are presented in Table 2. The best MAP and AUC are given by the multi-scale system using trees of depth 1. The statistical significance of the difference between this system and the best performing mono-scale one has been verified by a cross-validated paired t test [6]. This test indicated a significance of more than 99%.

Depth 1 trees yield better results for the multi-scale systems, but the choice of depth seems to have variable effects among mono-scale systems.

For comparison, in [2], the authors obtain a MAP and AUC of 0.54 and 0.73, respectively, on the same data and tags. But their system uses content-based and context-based information, whereas the one presented in this paper only relies on the audio content. However, the focus of this study is intentionally set on the methodological validation of the algorithm proposed, rather than achieving the best possible performance. Though, it shall be noticed that the ability of our new algorithm to handle data drawn on different scales makes it applicable to descriptors of different semantic levels, especially semantic information that may be valid at a smaller scale than the entire song (type of instrument, tempo, *etc.*). This very kind of data fusion will be explored in future works.

5. CONCLUSION

We proposed a new multi-scale fusion system for classification that is designed to be convenient for fusing heterogeneous features, both in terms of content description and scale. Fusion is done thanks to an adapted boosting algorithm using decision trees.

In this study, we focused on validating the ability of the proposed system to conveniently fuse features expressed at different scales. We experimented two classification tasks and the results show that the multi-scale system is the best one. Future work will study the ability of the system to fuse features that are describing different aspects of the musical pieces of interest, both in terms of content and scale.

6. REFERENCES

[1] L. Barrington, D. Turnbull, M. Yazdani, and G. Lanckriet. Combining audio content and social context for semantic music discovery. In *SIGIR*, pages 387–394, New York, NY, USA, 2009. ACM.

[2] L. Barrington, M. Yazdani, D. Turnbull, and G. Lanckriet. Combining feature kernels for semantic music retrieval. In *ISMIR*, pages 614–619, 2008.

[3] J. Bergstra and B. Kégl. Meta-features and adaboost for music classification. In *Machine Learning Journal*, 2006.

[4] T. Bertin-Mahieux, D. Eck, F. Maillet, and P. Lamere. Autotagger: a model for predicting social tags from

acoustic features on large music databases. *Journal of New Music Research*, 37(2):115–135, June 2008.

[5] T. Bertin-Mahieux, D. Eck, and M.I. Mandel. Automatic tagging of audio: The state-of-the-art. In Wenwu Wang, editor, *Machine Audition: Principles, Algorithms and Systems*. IGI Publishing, 2010.

[6] T.G. Dietterich. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 10(7), 1998.

[7] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 3 edition, 2009.

[8] J.L. Herlocker, J.A. Konstan, L.G. Terveen, and J.T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22:5–53, January 2004.

[9] C. Joder, S. Essid, and G. Richard. Temporal Integration for Audio Classification With Application to Musical Instrument Classification. *TASLP*, 17(1):174–186, 2009.

[10] M. Lardeur. *Robustesse des systèmes de classification automatique des signaux audio-fréquences aux effets sonores*. Master thesis, Université Pierre et Marie Curie, 2008.

[11] M.I. Mandel and D.P.W. Ellis. Multiple-instance learning for music information retrieval. In *ISMIR*, 2008.

[12] N. Mesgarani, M. Slaney, and S.A. Shamma. Discrimination of speech from nonspeech based on multiscale spectro-temporal Modulations. *TASLP*, 14(3):920–930, May 2006.

[13] G. Peeters and X. Rodet. Hierarchical Gaussian Tree with Inertia Ratio Maximization for the Classification of Large Musical Instruments Databases. In *Int. Conf. On Digital Audio Effects*, 2003.

[14] B.L. Sturm, M. Morvidone, and L. Daudet. Musical instrument identification using multiscale mel-frequency cepstral coefficients. In *EUSIPCO*, 2010.

[15] B.M. Taar Romeny. *Front-end vision and multi-scale image analysis*. Springer, 1st edition, 2003.

[16] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *TASLP*, 16(2):467–476, February 2008.

[17] G. Tzanetakis and P.R. Cook. Musical genre classification of audio signals. *Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[18] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, volume 1, pages 511–518, 2001.

[19] Y.H. Yang, Y.C. Lin, Y.F. Su, and H.H. Chen. A regression approach to music emotion recognition. *TASLP*, 16(2):448–457, 2008.