AN ADAPTIVE SYSTEM FOR MUSIC CLASSIFICATION AND TAGGING (MIREX 2009 SUBMISSION)

Juan José Burred and Geoffroy Peeters
IRCAM, Analysis/Synthesis Team - CNRS STMS
1, pl. Igor Stravinsky - 75004 Paris - France
{burred,peeters}@ircam.fr

ABSTRACT

This extended abstract concerns one of the two systems submitted by IRCAM for participation in the MIREX 2009 classification and tagging tasks. The system is adaptive and can handle both single-label classification tasks (genre, mood, artist) and multilabel tasks (tagging). Adaptability is attained by means of automatic feature and model selection, which are both embedded in the multiple-instance binary relevance learning of a Support Vector Machine. We propose a criterion function for SVM parameter selection that takes into account unbalanced sets and the effects of overfitting. The same algorithm, without any manual parameter adaptation, was submitted to all classification tasks. However, it was evaluated in two different configurations (also in all classification tasks) related to two different temporal modeling methods: in the first mode ("file") each track is represented by a single feature vector and in the second ("tw") texture windows of fixed length are computed, with a later temporal decision fusion.

1. INTRODUCTION

IRCAM has submitted two different systems for participation in all MIREX 2009 classification and tagging tasks. The first system, which we will call ircamclassification08 (denoted by GP in the MIREX results), is the same than last year's, and has been addressed in a separate abstract [1]. The system addressed here will be called ircamclassification09, and was denoted by BP in the results. This abstract contains a very brief description of the system. For a detailed presentation and discussion of the system, and for further experimental results, please see reference [2].

2. SYSTEM DESIGN

Ircamclassification09 is based on Support Vector Machines (SVM). Our proposal to attain adaptability involves embedding both feature and model selection in the multiple-instance binary learning needed for multiclass SVMs. Feature selection is based on the Inertia Ratio Maximization using Feature Space Projection (IRMFSP) algorithm [3]. Model selection involves searching for optimal SVM cost and kernel parameters by performing sub-cross-validation of the training database at each binary iteration, for which we propose to use a criterion function that takes into account overfitting and unbalanced sets.

An important characteristic of the proposed system is the binarization not only of the model training (which is needed for SVM anyway), but also of feature and model selection. Because this dramatically increases the overall model complexity (there are different features and model parameters for each binary instance), binarization of all learning stages is prone to overfitting. Thus, full binarization of feature and model selection should be accompanied by measures to mitigate overfitting in order for the system to gain in classification performance.

2.1 Feature extraction

A high adaptability calls for the extraction of a large number of audio features, that are to be subsequently selected automatically. All features are extracted on a short-term basis (Blackmann window of 60ms length and 20ms hop size), and include the following:

- **Basic spectral features.** Including spectral centroid, rolloff, flux, slope, skewness, kurtosis, etc.
- **Basic temporal features.** Autocorrelation and zero-crossings rate.
- **Perceptual features.** Loudness, specific loudness and a collection of spectral shape features (centroid, rolloff, flux, etc.) applied on a mel-warped spectrogram.
- **Harmonic features.** They measure the level of presence of sinusoidal components, as well as their overall spectral shape. They include noisiness, inharmonicity and harmonic spectral deviation.
- **MFCC.** 13 Mel Cepstral Coefficients are extracted, together with their first (Δ) and second ($\Delta\Delta$) derivatives.
- **Spectral Flatness Measure** and **Spectral Crest Measure.** They measure the flatness of the spectral envelope, and thus its noisiness.

- **Chroma coefficients.** Indicate the harmonic content by measuring the spectral energy in 12 frequency bands corresponding to the notes of the chromatic equal tempered scale.

An extracted short-time feature vector has a dimensionality of 280. To capture its dynamic behaviour, and to heavily reduce computational and storage requirements, a subsequent stage of temporal modeling is applied. In particular, the loudness-weighted mean and standard deviation of the features across a certain texture window (whose length is in the range of seconds) are extracted. This makes a total final dimensionality of 480.

Concerning temporal modeling, two different modes were evaluated:

1. In **file** mode (denoted by BP1 in the results), the texture window spans the whole track, independently of its length. The file mode is much more computationally efficient, but it might fail to capture some degree of dynamic feature behaviour.

2. In **tw** mode (denoted by BP2 in the results), texture windows of 4s length and 2s hop size are extracted. After classification, a decision fusion takes place. In single-label tasks, majority voting is used. In multi-label tasks, the track-wise label affinities are averaged, followed by a filtering according to a relevance threshold.

After extraction and temporal modeling, the axes of the feature space are centered and normalized by Inter-Quartile Range (IQR). The normalization parameters are extracted from the training set and used afterwards on the test set.

2.2 Binary feature and model selection

Our approach includes both feature and model selection to each one of the SVM binary repartitions. To that end, we use the *1-vs.-all* approach. Feature selection is based on the IRMFSP algorithm [3], which maximizes the Fisher discriminant (overall class separability) with an additional orthogonality constraint. A fixed number of 40 selected features was used.

The subsequent model selection stage involves searching for the optimal SVM parameters. Here, C-SVMs (*Slack variable*-SVMs) are used, since they attain a higher robustness against overfitting by allowing classification errors near the separation margin while learning. The cost of these errors is controlled by the factor c, which is one of the two parameters that need to be optimized. The other is the factor γ that controls the lobe width of the function used here as the kernel: Gaussian Radial Basis Function (G-RBF).

The most usual way of performing this parameter optimization is to perform a cross-validated exhaustive search in the (c, γ) grid, with classification accuracy as criterion function. In each fold of the validation, a parameter pair is selected and an SVM is trained and tested. The parameter pair corresponding to the highest obtained accuracy is selected. The cross-validation partitions are performed on the training set. To avoid confusion, we will call it sub-cross-validation (sCV).

Instead of using accuracy for parameter optimization, we propose the use of the following objective function:

$$\mathcal{F} = \mathrm{FMSR}(c_{ni}, \gamma_{nj}) \left(1 - \frac{\mathrm{S}(c_{ni}, \gamma_{nj})}{\mathrm{V}_n}\right), \qquad (1)$$

where FMSR is the F-Measure of the positive class in the current (n-th) binary sub-problem, S is number of support vectors found by the algorithm, V is the total number of training feature vectors, and $n = 1, \ldots, N$ is the binary sub-problem index. Such a function compensates unbalanced sets (by using the F-Measure instead of the accuracy) and takes into account overfitting (the number of support vectors found by the algorithm is a good indication of the complexity of the boundary).

2.3 C-SVM and probability estimates

After finding the optimal features and (c_n^*, γ_n^*) parameters, the n-th C-SVM with G-RBF as kernel is re-trained using the whole training set. In the classification phase, probability outputs are based on the pairwise coupling method [4].

3. IMPLEMENTATION DETAILS

The feature extraction module is based on the executable `ircamdescriptor`, which outputs the computed features in the binary SDIF format [5]. Its estimated runtime for every 30s of 22kHz, 16 bit mono WAV audio is of 3.3s in file mode and of 2s in tw mode, measured on an Intel Xeon 64 bit CPU at 2GHz and 8GB RAM. The required disk space per audio file is around 40kB in file mode (independently of the file size) and of 175kB for every 30s of 22kHz 16 bit WAV audio in tw mode.

SVM training and classification is performed by the `libsvm` library [6]. In file mode, total training runtimes per cross-validation fold depend heavily on the size of the database and number of classes. It can range from a few minutes for small databases with a small number of classes (5 to 10 classes) to around one hour for databases with a large number of classes (around 100). In texture window mode, the training algorithm is much more computationally demanding. Also, the size of the database has less effect on the total runtime. Even for a small database, a single cross-validation fold can take several hours (however, one fold should not take more than two hours).

4. MIREX 2009 RESULTS

The system was evaluated in all 4 single-label classification tasks and both multilabel (tagging) tasks. It should be noted that there were no task-specific configurations or parameter settings prior to submission: the same system with the same parameters was tested with all 6 databases.

(a) Genre (mixed)

CL2	73.33%
CL1	73.23%
GLR1	71.23%
BP1	**70.63%**
MTG5	70.44%
XZZ	69.36%
XLZZG	68.93%
VA1	68.84%
BP2	**68.51%**
LZG	68.29%
TTOS	67.89%
GT2	67.87%
VA2	67.39%
SS	66.60%
HW1	65.99%
HW2	65.31%
GT1	65.10%
MTG1	64.79%
HNOS1	64.47%
HNOS3	64.34%
GP	64.24%
MTG3	64.06%
MTG4	64.00%
RK1	61.41%
ANO	60.50%
GLR2	60.14%
RCJ4	50.99%
HNOS4	45.16%
RCJ3	37.71%
RCJ1	32.50%
HNOS2	20.90%

(b) Genre (latin)

CL1	74.66%
CL2	73.58%
BP1	**67.31%**
SS	64.69%
BP2	**63.52%**
MTG6	63.16%
GLR1	62.79%
GP	62.63%
MTG2	62.39%
MTG1	61.68%
MTG5	61.14%
VA1	58.37%
RK1	57.11%
HNOS1	56.32%
HNOS3	56.22%
LZG	55.96%
XLZZG	55.25%
XZZ	55.25%
RCJ4	55.22%
HW1	54.72%
VA2	54.49%
TTOS	53.70%
GT2	52.82%
RCJ2	52.43%
HW2	52.28%
GLR2	49.84%
GT1	49.75%
MTG4	47.79%
RCJ3	46.78%
MTG3	45.80%
RCJ1	38.93%
ANO	38.87%
HNOS4	30.05%

(c) Classical composer

MTG2	62.05%
CL1	60.97%
CL2	60.03%
XZZ	57.18%
HW1	56.35%
BP1	**55.66%**
GLR1	55.34%
BP2	**54.76%**
MTG1	54.73%
LZG	54.40%
VA1	53.57%
VA2	53.57%
XLZZG	53.54%
HW2	53.10%
SS	52.56%
GT2	51.48%
MTG6	50.36%
MTG5	49.75%
GP	48.85%
RK1	48.41%
MTG4	48.20%
MTG3	48.12%
GLR2	45.92%
TTOS	44.37%
GT1	43.69%
HNOS1	43.33%
HNOS3	42.24%
ANO	41.77%
HNOS4	29.04%
HNOS2	15.84%

(d) Mood

CL1	65.67%
CL2	65.50%
GP	63.67%
MTG5	62.83%
HW2	61.67%
LZG	61.67%
HW1	61.33%
GLR1	60.83%
FCY1	60.33%
VA2	60.17%
XZZ	60.00%
MTG3	59.83%
BP2	**59.67%**
MTG6	59.50%
GT1	59.33%
MTG4	59.33%
VA1	59.33%
SS	58.83%
HNOS1	58.67%
HNOS3	58.67%
FCY2	58.33%
BP1	**58.17%**
MTG1	57.67%
MTG2	57.50%
XLZZG	57.00%
GT2	56.83%
TAOS	56.83%
RK1	53.17%
GLR2	53.00%
HNOS4	51.17%
ANO	50.67%
RK2	41.33%
HNOS2	34.67%

Table 1. Results of the MIREX 2009 single-label classification tasks (mean classification accuracy in %).

4.1 Single-label results

Table 1 shows the results in terms of mean classification accuracy for all 4 single-label classification tasks. The system has shown good performance in the 10-class mixed genre task (4th best out of 31 algorithms), in the 10-class latin genre task (3rd best out of 33 algorithms) and in the 11-class classical composer task (6th best out of 30 algorithms). A noteworthy result in these 3 cases is that the system performed better in file mode (BP1), with a single feature vector representing each track, rather than using texture windows with temporal decision fusion.

In the 5-class mood task, the results show a different behavior of the system, both in terms of performance (which is more moderate, ranking 13th out of 33) and in terms of a better performance of the texture window approach (BP2). Also, the closeness of the accuracy results for most of the participating algorithms is especially remarkable in this case.

4.2 Multilabel results

As shown in Table 2, in terms of average tag-level F-Measure the system ranked 4th with the MajorMiner dataset (43 labels) and 5th with the mood dataset (18 labels).

(a) MajorMiner

LWW2	0.3107
GT2	0.2933
GT1	0.2900
BP2	**0.2899**
LWW1	0.2890
BP1	**0.2767**
CC4	0.2626
CC2	0.2414
CC1	0.2093
CC3	0.1705
HBC	0.0443
GP	0.0122

(b) Mood

LWW2	0.2195
GT1	0.2114
GT2	0.2088
LWW1	0.2037
BP1	**0.1949**
BP2	**0.1926**
CC4	0.1833
CC2	0.1798
CC1	0.1723
CC3	0.1471
GP	0.0840
HCB	0.0632

Table 2. Results of the MIREX 2009 multilabel classification tasks (average tag F-Measure).

5. ACKNOWLEDGEMENTS

This work was realized as part of the Quaero Programme [1], funded by OSEO, the French State agency for innovation. We would like to thank Carmine Emanuele Cella and Frédéric Cornu for their work as developers of the feature extraction module.

[1] http://www.quaero.org

6. REFERENCES

[1] G. Peeters. MIREX-09 music mood, mixed-genre, latin-genre and classical composer classification tasks: ircamclassification08 submission. In *Music Information Retrieval Evaluation Exchange (MIREX)*, October 2009.

[2] J.J. Burred and G. Peeters. An adaptive system for music classification and tagging. In *Proc. International Workshop on Learning the Semantics of Audio Signals (LSAS)*, Graz, Austria, December 2009.

[3] G. Peeters. Automatic classification of large musical

instrument databases using hierarchical classifiers with inertia ratio maximization. In *Proc. 115th Convention of the Audio Engineering Society*, New York, USA, October 2003.

[4] T.-F. Wu, C.-J. Lin, and R. C. Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5:975–1005, 2004.

[5] J.J. Burred, C.E. Cella, G. Peeters, A. Röbel, and D. Schwarz. Using the SDIF Sound Description Interchange Format for audio features. In *Proc. International Conference on Music Information Retrieval (ISMIR)*, Philadelphia, USA, September, 2008.

[6] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.