An Adaptive System for Music Classification and Tagging

Juan José Burred and Geoffroy Peeters

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

Abstract. We present a system that can learn effective classification models from music databases of very different characteristics, including both single-label collections indexed by genre or artist and multilabel databases of musical mood and instrumentation, where multiple tags can be applied to each track. Adaptability is attained by means of automatic feature and model selection, both embedded in the multiple-instance binary relevance learning of a Support Vector Machine. We discuss strategies for compensating overfitting and unbalanced training sets.

1 Introduction

Recent developments in Music Information Retrieval technologies have followed the trend of shifting from the classical single-label, single-criterion model of classification towards a multi-label, multi-criteria paradigm. The musicological difficulties of hard-assigning musical tracks to fixed categories such as genres limit the usefulness of a system in practical use, even if it is able to reach satisfying performances in the laboratory for particular tasks. This is not only due to technical challenges of the pattern recognition algorithms involved, but mainly because of the system failing to meet users' expectations concerning their own understanding of the musical categories.

It should be noted that multi-criteria and multi-label are two independent concepts. The label multiplicity (single or multiple) refers to the number of labels the system can output *per track*. The criteria refer to the categories described by the labels (genre, mood, instrumentation, etc.). A multi-criteria system can be single- or multi-label, however the latter case is more common.

Multilabel music classification is also known as music *tagging*, and has only recently started to gain interest with the advent of popular online music services based on social networking and collaborative-filtering-based recommendation. Tagging with multiple labels avoids the difficulty and inaccuracy of describing a whole musical track with a single genre, mood, instrumentation or any other possible label.

A system capable of working with different criteria needs to be adaptive, in order to accommodate itself to databases of potentially very different number of classes, number of audio files and qualities of annotation. Thus, the key machine

learning concepts involved are automatic feature and model selection. On the other hand, in benefit of general applicability, it must avoid a too high modeling accuracy on the training set, i.e., it must avoid *overfitting*. The most demanding goal of a classification application is not accuracy, but generality. Using highly complex decision functions that work well on the training set might perform poorly when the system is subjected to cross-validation. It is therefore crucial to find a trade-off between system adaptability and overfitting.

Only relatively recent works have addressed adaptability and multilabel capabilities of music classification systems. An example is the system proposed in [1], where adaptability is achieved through automatic feature selection and a Gaussian Mixture Model (GMM)-based classifier is tested in speech/music segregation and genre classification tasks. In [2], multilabel classification is applied to music and sound effect databases by using label-level GMM distributions learnt with a hierarchical Expectation-Maximization algorithm. In [3], the Random k-Labelsets (RAKEL) algorithm for multilabel classification is applied to music mood detection. RAKEL has the particularity to handle the multilabel problem *all at once*, instead of decomposing the problem into a set of sub-problems. The first MIREX contest for music tagging took place in the 2008 edition [4].

We present a system that takes into account the demands of adaptability and subject it to extensive evaluation with two single-label databases (music genre and artist detection) and two multilabel databases (mood and instrumentation). Classification is based on Support Vector Machines (SVM). Our proposal to attain adaptability involves embedding 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 [5]. 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. The handling of unbalanced sets is crucial for tagging applications, and for single-label applications with a high number of classes, as will be discussed more in detail.

An important characteristic of the proposed system is the binarization[1] 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, binarization should be accompanied by measures to mitigate overfitting in order for the system to gain in classification performance. Computational requirements also increase, but the separation in binary instances allows easy parallelization.

In the following section we explain in detail the different components of the system. Sect. 3 briefly introduces the four music databases used in the evaluation experiments detailed in Sect. 5, and Sect. 4 emphasizes on the two different

[1] In this context, *binarization* is the conversion of a multiclass problem into a set of 2-class sub-problems.

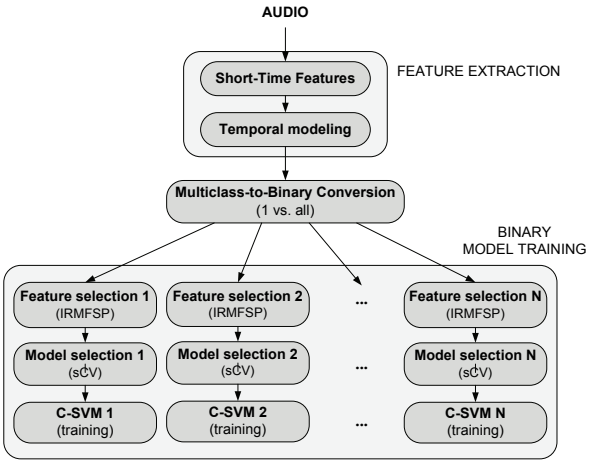

Fig. 1. Overview of the system in training.

evaluation approaches that are needed for the single-label and the multilabel cases.

2 System modules

Overviews of the system in training and classification modes are shown in Fig. 1 and Fig. 2, respectively. Note that the only modules that are specific to either single-label or multilabel classification are the decision fusion modules in the classification subsystem. All the others are valid for both annotation modes.

2.1 Feature extraction

A high adaptability calls for the extraction of a large number of audio features (most of them described in detail in [6]), that are to be subsequently selected automatically. All features are extracted on a short-term basis, 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.

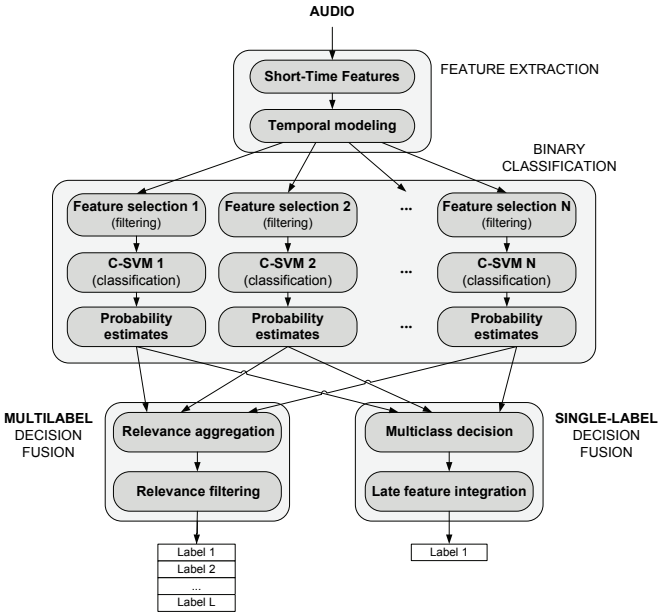

Fig. 2. Overview of the system in classification.

- **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.

After extraction and temporal modeling, the axes of the feature space are centered and normalized by Inter-Quartile Range (IQR). The normalization pa-

rameters are extracted from the training set and used afterwards on the test set.

2.2 Binarization

The conversion of a multiclass[2] problem into a set of 2-class sub-problems appears naturally in the context of SVM-based classifiers. Most multiclass SVM implementations operate by a series of binary repartitions of the database prior to actual binary SVM training, followed by some voting or decision scheme. Usually, the database repartition is embedded into the SVM algorithm and thus other learning stages such as feature extraction and model selection are kept out of the binarization and performed in a multiclass context. In such a situation, the found optimal features and model parameters are the same for all the subsequent pairwise SVM classifications.

We use here an alternative approach consisting in including both feature and model selection to each one of the binary repartitions. This has the potential of improving classification performance if the optimal pairwise separation boundaries between classes are highly dissimilar to each other. For example, we might need a completely different set of features, and a different degree of nonlinearity in the kernel mapping, when separating jazz from blues, than for separating jazz from hard rock. A higher boundary nonlinearity will probably be needed in the first case.

We use the *1-vs.-all* approach to binarization, in which a multiclass problem of C classes is subdivided as a set of C binary sub-problems. In the binary sub-problems, the positive class is the class under consideration, and the negative class is made up of the rest of the training database. Another popular approach is *1-vs.-1* binarization, in which the number of subproblems is $C(C-1)/2$. In the case of traditional SVM learning, *1-vs.-all* and *1-vs.-1* have been reported as having similar classification and computational performances [7] (in the latter case, the higher number of sub-problems is compensated by the lower number of feature vectors in the negative classes). In our case, however, the *1-vs.-1* case would be much more computationally demanding, since also feature and model selection are run in each binary instance, and their performance is far less related to the number of feature vectors in the corresponding classes.

2.3 Binary feature and model selection

Feature selection is based on the IRMFSP algorithm [5], which maximizes the Fisher discriminant (overall class separability) with an additional orthogonality constraint. A subsequent dimensionality reduction step based on Linear Discriminant Analysis (LDA) was tested in preliminary experiments, but was confirmed to be inappropriate in a binary context, since it projects all the selected features into a single dimension, a too coarse simplification.

[2] *Multiclass* (more than 2 classes in the training set) should not be confused with *multilabel* (more than 1 class can be assigned assigned to one track).

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. Note that the cross-validation partitions are actually performed on the training set, not in the whole evaluation database (which would amount to learning from the test set). To avoid confusion, we will call it sub-cross-validation (sCV).

Using accuracy as criterion can be however inefficient in binary sub-problems arising from a *1-vs.-all* binarization, since the two involved classes will almost certainly be unbalanced in the number of feature vectors (the negative class will contain many more vectors than the positive class). Thus, a high overall accuracy will be obtained even if very few (or even no) true positives are detected, and the selected parameters will be unoptimal in the final evaluation tests. In such cases, a more appropriate alternative is to use the *F-Measure* (FMSR) of the positive class[3], which is the harmonic mean of the recall (RCL) and precision (PRC) of that class:

$$\mathrm{RCL} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{TN}}, \quad \mathrm{PRC} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \quad \mathrm{FMSR} = \frac{2 \cdot \mathrm{PRC} \cdot \mathrm{RCL}}{\mathrm{PRC} + \mathrm{RCL}}, \qquad (1)$$

where TP are the true positives, TN are the true negatives, FP are the false positives and FN are the false negatives.

In addition to the F-Measure, we include an additional term in the objective function, which measures the proportion of support vectors found. The number of support vectors is a good indication of the degree of overfitting. The support vectors are the training vectors that define the optimal separation boundary, as found by the SVM training optimization. Thus, proportionally fewer support vectors imply a less complex decision function, and therefore a (likely) more generalizable model. The final parameter optimization problem based on the proposed function takes the form:

$$(c_n^*, \gamma_n^*) = \underset{c_{ni}, \gamma_{nj}}{\operatorname{argmax}} \left\{ \mathrm{FMSR}(c_{ni}, \gamma_{nj}) \left(1 - \frac{\mathrm{S}(c_{ni}, \gamma_{nj})}{\mathrm{V}_n} \right) \right\}, \qquad (2)$$

where S is the number of support vectors found by the algorithm, V_n is the total number of training feature vectors in the current binary sub-problem, and $n =$

[3] Note that, in a binary problem, "F-Measure of the positive class" is synonymous with just "F-Measure". This is not the case in multiclass problems, where F-Measure, as well as recall and precision, are always defined in relation to a specific class.

$1, \ldots, N$ is the binary sub-problem index. It should be noted that the size of the grid defined by (c_{ni}, γ_{nj}) has a huge impact on computational time. Therefore, its resolution is automatically chosen based on the size of the training database. On the other hand, grids with too high a resolution attain lower performance due to overfitting.

2.4 C-SVM and probability estimates

After finding the optimal features and (c_n^*, γ_n^*) parameters, the n-th C-SVM[4] with G-RBF as kernel is re-trained using the whole training set. The formulation of the training for the n-th binary instance is thus the following:

$$\min_{\mathbf{w}_n, \boldsymbol{\xi}_n, b_n} J(\mathbf{w}_n, \xi_n) = \frac{1}{2}\|\mathbf{w}_n\|^2 + c_n^* \sum_{i=1}^{V_n} \xi_{ni}, \tag{3}$$

$$\text{subject to} \quad (\mathbf{w}_n^T \phi(\mathbf{x}_i) + b_n) \geq 1 - \xi_{ni}, \quad \text{if} \quad y_i = n,$$

$$(\mathbf{w}_n^T \phi(\mathbf{x}_i) + b_n) \leq -1 + \xi_{ni}, \quad \text{if} \quad y_i \neq n,$$

$$\text{and} \quad \xi_{ni} \geq 0, \quad \forall i, n,$$

where $\mathbf{w}_n$ is the normal vector defining the n-th separating hyperplane, ξ_{ni} are the slack variables associated to the n-th sub-problem, $\phi(\cdot)$ is the mapping function associated to the kernel and b_n are the hyperplane offsets.

In the classification phase, after filtering out the selected descriptors, classification on the m-th trained SVM is done based on a decision function of the form $\mathbf{w}_n^T \phi(\mathbf{x}_i) + b_n$. For classification and retrieval applications, a more convenient output is the probability of a classified vector to belong to the different classes. This allows later probabilistic temporal integration and computation of class or tag relevances. Here, probability estimation is based on the pairwise coupling method proposed by Wu *et al.* [9].

2.5 Decision fusion for single-label tasks

In the single-label case, the single most probable class for the whole music track has to be selected out of the set of binary classifications, and also out of the set of classifications corresponding to the temporal sequence of the texture windows of that piece. Decision fusion is thus implemented in two phases: first, the most probable of the N binary positive classes is selected for each texture window, followed by a majority voting of all the detected classes for all the texture windows of the track (this is labeled as *late feature integration* in Fig. 2).

[4] We use the `libsvm` library [8] as SVM implementation.

2.6 Decision fusion for multilabel tasks

Decision fusion has a different goal in multilabel tasks. Instead of choosing a winning class per track, the decision involves selecting a subset of $L < N$ labels that are judged as relevant to the track. Note that the set of output labels can vary in size for different tracks, and might even be empty if no relevant label is found. This is in contrast to the single-label scenario, in which a class has to be always assigned (and even if a vector of output probabilities is given in a single-label problem, its size is always fixed and equal to the total number of classes). Thus, the size of the output label set is an additional parameter exclusive to multilabel tasks, that needs to be carefully optimized.

The sequence of multiclass and temporal decision fusion must now be inverted: first, the probabilities of all positive classes across the sequence of all texture windows for a given track are first averaged (*relevance aggregation*), followed by the filtering of the most relevant tags by means of the relevance threshold (*relevance filtering*). An adequate relevance threshold is crucial for a satisfactory balance between label-based precision and recall in evaluation. The threshold has been optimized by cross-validation on the whole system.

3 Databases

Four different annotated music databases have been prepared for the evaluations. They follow different annotation criteria and have very different class distributions and populations.

3.1 Single-label databases

- **Genre.** The publicly available ISMIR 2004 music genre database[5] has been used. It contains 1422 copyright-free mp3 tracks (128 kbps, 44.1 kHz) organized into 6 genres (see Table 1). Approximately half of the files are separated as a "training database" and the rest as a "development database", and several algorithms in the literature have been evaluated by testing the development database against the training database. In order to allow a direct comparison, we chose to follow the same evaluation method, rather than using k-fold cross-validation.
- **Artist.** A database of 3150 MP3 clips of 30 seconds extracted around the center of each song (128 kbps, 32 kHz) has been compiled, containing tracks from 105 pop/rock artists (30 tracks per artist). This database has been designed to resemble the (not publicly available) MIREX 2008 [4] artist detection database in size and proportions, but contains different artists and audio files. The artist labels originate from the ID3 metadata of the MP3 files.

[5] `http://ismir2004.ismir.net/genre_contest/index.htm`

Database	**Genre**		**Artist**
#files	1422		3150
#classes	6		105
#files per class	classical	604	30 files per artist (mostly pop/rock)
	electronic	229	
	jazz/blues	52	
	metal/punk	90	
	rock/pop	203	
	world	244	
Annotation	Metadata		Metadata
Evaluation	1-fold cross-database		3-fold cross-validation

Table 1. Single-label databases used for evaluation.

3.2 Multilabel databases

The database characteristics and classes for the multilabel case are shown in Table 2. The figure contains an additional measure: the *label cardinality*, i.e., the average number of annotated labels per track.

- **Mood.** 193 mp3 files (128 kbps, 32 kHz) have been manually annotated with labels reflecting mood or emotion characteristics. Multiple labels per file are allowed (but not required). Each file has been annotated 3 times in order to allow an assessment of label relevance by measuring the agreement between annotators. From a bigger initial set of labels, only those labels for which 2 of the three annotators agreed, and that appear in at least 6 tracks, were kept. Since this annotation process is much more costly than in the single-label case, the size of the database is smaller.
- **Instrumentation.** 252 mp3 files (128 kbps, 32 kHz) have been manually annotated with a variety of criteria related to instrumentation (lead vocal type, drum kit type, production style, etc.). Again, 3 annotations per file were done, and only the labels with an agreement of 2 or 3 were kept. Note that the label cardinality is twice as high as in the mood case, and that the classes are better populated.

4 Evaluation measures

It is possible to define a unified evaluation framework valid both for single-lable and multilabel scenarios by reinterpreting in each case the meaning of TP, TN, FP and FN, or even more easily, by reinterpreting the meaning of the *answer set* $\mathcal{A}$ and the *relevant set* $\mathcal{R}$. In terms of general Information Retrieval, the answer set is the set containing all items output by the algorithm in response to a certain query, and the relevant set contains all items annotated as being relevant to a particular query (i.e., the *ground truth*). The class-wise positive and negative scores are then defined for class n as:

$$\mathrm{TP}_n = |\mathcal{R}_n \cap \mathcal{A}_n| \tag{4}$$

Database	**Mood**		**Instrumentation**	
#files	193		252	
#labels	15		19	
label cardinality	2.06		3.99	
#files per label	angry/aggressive	13	background vocals	64
	bizarre/weird	6	drum kit: techno	22
	calming/soothing	50	drum kit: electronic 80's	13
	cheerful/festive	46	drum kit: heavy rock/metal	21
	contrasted	11	drum kit: jazz/country/soul	51
	enchanting/magical	6	drum kit: light pop/rock	97
	grandiloquent	13	drum kit: urban / R'n'B / rap	24
	laid-back/mellow	20	guitar solo	27
	mechanical/robotic	9	piece based on distorted guitars	43
	playful	40	simple presence of distorted guitars	17
	positive/happy	52	instrumentation archetype: electronic	32
	powerful/strong	49	instrumentation archetype: vocal and acc.	10
	romantic/passionate	34	instrumentation archetype: pop/rock	164
	sad/melancholic/doleful	9	lead vocal part	154
	sophisticated/elegant	41	no melodic reference	14
			not a lead vocal part	48
			production: heavily produced in studio	50
			production: produced with acoustic instruments	93
			production: transparent	60
Annotation	3 manual annotations per file		3 manual annotations per file	
Evaluation	3-fold cross-validation		3-fold cross-validation	

Table 2. Multilabel databases used for evaluation.

$$\mathrm{TN}_n = |\mathcal{S} - (\mathcal{R}_n \cup \mathcal{A}_n)| \tag{5}$$

$$\mathrm{FP}_n = |\mathcal{A}_n - (\mathcal{R}_n \cap \mathcal{A}_n)| \tag{6}$$

$$\mathrm{FN}_n = |\mathcal{R}_n - (\mathcal{R}_n \cap \mathcal{A}_n)|, \tag{7}$$

where $\mathcal{S}$ is the set of all items, $|\cdot|$ denotes the number of elements in a set and the $-$ denotes set difference. There is one interpretation of the sets for the single-label case, and two possible interpretations for the multilabel case, which are the following:

- **Single-label:** $\mathcal{S}$ is the set of all tracks in the test partition of the database, $\mathcal{A}_n$ is the set of test tracks assigned by the algorithm to class n, and $\mathcal{R}_n$ is the set of test tracks annotated in the ground truth as belonging to class n.
- **Multilabel label-based measures:** $\mathcal{S}$ is the set of all tracks in the test partition of the database, $\mathcal{A}_n$ is the set of test tracks assigned by the algorithm to label n, and $\mathcal{R}_n$ is the set of test tracks which include label n in the ground truth annotation.
- **Multilabel track-based measures:** $\mathcal{S}$ is the set of all labels in the ground truth (the *dictionary*), $\mathcal{A}_n$ is the set of labels assigned by the algorithm to track t, and $\mathcal{R}_n$ is the set of all labels annotated for track t in the ground truth.

Multilabel track-based measures can be misleading about the generalized performance of the system. They can lead to artificially good results if a system

is good at predicting a few well-populated labels (such as "pop/rock" instrumentation) and bad at predicting rare or more specific labels (such as "electronic 80's drum kit") [2]. For a more generalized performance indication that ensures that even the rare labels are well classified, label-based measures should be used, as we did in the experiments outlined in the next section.

Once the positive/negative scores values are computed, the class-wise evaluation measures of RCL, PRC and FMSR are computed as in Eq. 1. In addition, in the single-label case a popular measure is the accuracy (ACC), which is simply the percentage of tracks correctly classified. In spite of its popularity, this measure can be misleading with unbalanced datasets, as has been argued before. The F-Measure should be considered as the most robust and informative measure of performance in both single- and multilabel cases.

5 Experimental results

The goal of the experiments was not only to test the performance of the system in the individual tasks, but also its adaptability, without manual changes, between databases of very different characteristics. In this respect, we emphasize that the parameters that are not automatically optimized by the system (sound analysis parameters, number of selected features, multilabel relevance threshold, etc.) remained unchanged between all experiment runs with all four single- and multilabel databases. For each evaluation configuration, the system was launched with each one of the databases with no manual parameter tuning or changing between runs.

For the short-term feature extraction[6], a Blackman window of 60ms length and a hop size of 20ms was used. For the single-label experiments, two different temporal modeling methods were tested: the first ("file") takes the whole track length as a single texture window, and thus each track is represented by a single feature vector. In the second mode ("tw"), a texture window of a fixed length of 4s and a hop size of 2s was set. The file mode is much more computationally efficient, but it might fail to capture some degree of dynamic feature behaviour. For reasons of computational demands, multi-label databases were only tested in file mode. After assessing the performance in preliminary tests, a fixed number of 40 selected features was set for all final experiments.

The results for the single-label experiments are shown in Table 3, and for the multilabel experiments in Table 4. All measures are averaged across classes and, in the artist, mood and instrumentation cases, across the 3 folds of the cross-validation. The "multiclass" configuration denotes the usual approach of performing feature and model selection on the multiclass dataset, outside of the binarization. Thus, the features selected by the IRMFSP algorithm and the parameters selected by multiclass sCV (with accuracy as a criterion function) are common for all n SVMs. Note that this does not apply to the multilabel case, in which a binarization of the whole training has to be carried out anyway.

[6] All MP3 files were decoded into WAV before processing.

		SINGLE-LABEL							
	temp.	**Genre**				**Artist**			
Configuration	mod.	ACC	RCL	PRC	FMSR	ACC	RCL	PRC	FMSR
Multiclass	file	83.95	78.84	80.31	79.08	28.44	28.44	34.71	27.12
BFS - sCV(acc)	file	84.91	79.34	80.98	79.95	**45.71**	**45.71**	**47.98**	**45.04**
BFS - sCV(f+sv)	file	85.32	79.47	81.32	80.15	45.40	45.40	47.58	44.60
Multiclass	tw	87.11	81.26	85.20	83.03	41.59	41.59	40.76	39.63
BFS - sCV(acc)	tw	87.24	81.79	86.38	83.79	44.35	44.35	46.94	43.67
BFS - sCV(f+sv)	tw	**88.07**	**82.80**	**86.95**	**84.62**	43.33	43.33	45.90	42.53

Table 3. Results for the single-label databases. All measures are averaged across classes and (in the case of the artist database) across cross-validation folds.

		MULTILABEL					
	temp.	**Mood**			**Instrumentation**		
Configuration	mod.	RCL	PRC	FMSR	RCL	PRC	FMSR
BFS - sCV(acc)	file	**78.59**	22.01	32.03	**85.52**	34.24	46.28
BFS - sCV(f+sv)	file	59.85	**33.61**	**40.23**	74.59	**42.38**	**52.32**

Table 4. Results for the multilabel databases. All measures are label-based and averaged across labels and across cross-validation folds.

The row indicated as "BFS-sCV(acc)" (Binary Feature Selection (BFS) and binary sub-cross-validation (sCV) based on accuracy (acc)) corresponds to the full binarization of IRMFSP feature selection and sCV model selection. Since the criterion function (acc) is still the same than in the multiclass case, the results on this row show the independent effect of binarization. The performace has been improved in all cases (in terms of F-Measure), both with file and texture-window temporal modeling. The improvement is slight in the genre case (6 classes), and more important in the artist case (105 classes). This shows that using the same parameters for a large set of decision boundaries is too broad a simplification, and that binarization is convenient in such cases.

The "BFS-sCV(f+sv)" denotes again full binarization, but with the accuracy criterion function replaced by the function proposed in Eq. 2, which takes into account unbalancing and overfitting. The use of this function improves the F-Measure in all but the artist database. Note in particular that the improved performance for the multilabel databases is due to a better balance between precision and recall. A possible reason why the performance is similar but not better with the 105-class artist database might be the extreme unbalancing of the binary sub-problems, which would need further compensation. A possibility we will investigate is to learn two different cost coefficients in each sub-problem, one for the positive class (c^*_{+n}) and one for the negative class (c^*_{-n}).

The genre results are directly comparable to previous approaches, since they are based on a very similar database. The class-averaged recall reported in [1] was of 78.7% with the same training and development databases and the best recall obtained in the ISMIR 2004 genre contest was of 78.78%, in this case with the development database replaced by a non-public evaluation database

of the same characteristics and proportions. In comparison, the present system obtained a mean recall of 82.80%.

The system was also submitted for participation in the MIREX 2009 classification and tagging tasks. For details on the evaluation results, see [10].

6 Conclusions

Adaptability of the presented system has been achieved through the use of automatic feature selection from a large set of features, through the use of SVM cost and kernel parameter selection by sub-cross-validation, and by allowing both single-label and multilabel modes of operation. General performance can be significantly improved by binarizing all stages of the training process, not only SVM training (as is usually the case), but also of feature and model selection. Performance can be further optimized by using a model selection criteron function that takes into account training set unbalancing and overfitting. The best performances obtained with the same configuration (full binarization with "f+sv" criterion function) were of 88.07% accuracy and 84.62% F-measure for a 6-class single-label genre database, of 45.40% accuracy and 44.60% F-Measure for a 105-class single-label artist database, of 40.23% label-based F-Measure for a 15-label multilabel mood database, and of 52.32% label-based F-Measure for a 19-label multilabel and multi-criteria instrumentation database.

A possibility towards further adaptability would be to automatically choose one of the configurations shown in Tables 3 and 4 depending on the class population and database size. But for a robust choice, this will probably need further extensive testing, and the use of bigger databases, especially in the multilabel case. Another future direction of research will be to explore more informative temporal modeling methods.

7 Acknowledgements

This work was realized as part of the Quaero Programme[7], funded by OSEO, the French State agency for innovation. The multilabel annotations were performed by Emmanuel Deruty, Maxence Riffault and Jean-François Rousse. We also thank Carmine Emanuele Cella and Frédéric Cornu for their work as developers of the feature extraction module.

References

1. G. Peeters, "A generic system for audio indexing: Application to speech/music segmentation and music genre recognition," in *Proc. International Conference on Digital Audio Effects (DAFX)*, Bordeaux, France, September 2007.

[7] `http://www.quaero.org`

2. D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, "Semantic annotation and retrieval of music and sound effects," *IEEE Transactions on Acoustics, Speech, and Language Processing*, vol. 16, no. 2, pp. 467–476, 2008.
3. K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, "Multi-label classification of music into emotions," in *Proc. International Conference on Music Information Retrieval (ISMIR)*, Philadelphia, USA, September 2008.
4. MIREX, "Music Information Retrieval Evaluation eXchange," 2008, http://www.music-ir.org/mirex/2008/.
5. 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.
6. G. Peeters, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," in *CUIDADO I.S.T. Project Report*, 2004.
7. C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.
8. 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.
9. T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research*, vol. 5, pp. 975–1005, 2004.
10. MIREX, "Music Information Retrieval Evaluation eXchange," 2009, http://www.music-ir.org/mirex/2009/.