BASS TRACK SELECTION IN MIDI FILES AND MULTIMODAL IMPLICATIONS TO MELODY

Octavio Vicente and José M. Iñesta
University of Alicante, Alicante, Spain

Keywords: Multimodal pattern recognition, Music information retrieval, Bass and melody, Symbolic data.

Abstract: Standard MIDI files consist of a number of tracks containing information that can be considered as a symbolic representation of music. Usually each track represents an instrument or voice in a music piece. The goal for this work is to identify the track that contains the bass line. This information is very relevant for a number of tasks like rhythm analysis or harmonic segmentation, among others. It is not easy since a bass line can be performed by very different kinds of instruments. We have approached this problem by using statistical features from the symbolic representation of music and a random forest classifier. The first experiment was to classify a track as bass or non-bass. Then we have tried to select the correct bass track in a multi-track MIDI file. Eventually, we have studied the issue of how different sources of information can help in this latter task. In particular, we have analyzed the interactions between bass and melody information. Yielded results were very accurate and melody track identification was significantly improved when using this kind of multimodal help.

1 INTRODUCTION

Nowadays it is possible to access to large on-line music collections, and this music could be in some audio or symbolic formats. MIDI is one of the most utilized systems to encode symbolic music, but it is also possible to find other formats for digital scores, such as MusicXML, ESAC or MuseData, among others.

Therefore it is necessary to manage huge music databases organizing them by genre, mood, author of the piece, or by the instruments played in them, for example. On the other hand, it is also necessary to index these databases in order to find music pieces.

The melody line in a music work often gets relevant information about the whole piece, and usually it is possible to determine the genre or mood from it. It is also possible to organize the database using descriptors or other information from the melody line. It is expected that the user will query the database using a fragment of the melody, so to extract it from the file is a valuable information for building the targets for those queries. It is also possible to find similar songs to a gien one using the melody line. Therefore, Music Information Retrieval (MIR) researchers have shown an increased interest in extracting the melody part from multitrack files.

Although melody line offers valuable information, it could not be enough in order to get some particular informations from it. For instance, the bass line of a music piece usually conveys relevant features, as the harmonicity or the rhythm, that can be utilized in a number of MIR tasks, like chord segmentation, cover version recognition, of for classifying the genre or the mood for that music piece.

For example, a given classical melody can have jazz or rock versions, keeping the melody but changing instrumentation and rhythm. Hence, the melody for all of them can be roughly the same and therefore it is not enough to determine the genre for each version. On the other hand, it is commonly accepted that it is harmony (chord sequence) what is kept among different cover versions of a song. Therefore, the information retrieved from the bass line of a multitrack symbolic file can be used to get informations that the melody line can not provide.

Standard MIDI files consist of a number of tracks containing information that can be considered as a symbolic representation of music. Usually each track represents an instrument or voice in a music piece.

This works aims first to take a methodology already known for melody part selection and adapt it to select the bass track from a multi track MIDI file. Then, we will explore how the bass line characterization in terms of probabilities can help in the melody part selection task under a multimodal approach.

This work is based on symbolic data description

Vicente O. and Iñesta J..
BASS TRACK SELECTION IN MIDI FILES AND MULTIMODAL IMPLICATIONS TO MELODY.
DOI: 10.5220/0003843404490458
In *Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods* (IATMLRP-2012), pages 449-458
ISBN: 978-989-8425-98-0

methodologies in a MIR context. Although MIDI files have been used here, the features for the classifier implementation are based on statistical descriptors from the notes contained in the track. Therefore it is possible to change the standard MIDI files music source by any other symbolic format, like MusicXML for example, just by changing the feature extraction front-end in the implemented application.

2 RELATED WORKS

A considerable amount of works have been published in MIR using machine learning and pattern recognition techniques for musical content description, both in the audio and symbolic domain.

Works in the audio domain aim to get information from songs to classify them by genre or mood, or for extracting fingerprints that characterize them for indexing, retrieval or content-based playlist generation. Audio files contain a wave, therefore the informations extracted from these files are in the frequency and time domains. Music transcription has proven to be valuable to get a symbolic representation of the sounds played in the songs with a number of MIR task applications (Lidy et al., 2007). On the other hand, source separation researches aim to get the different voices or instrument lines from the songs, like in (Kim and Choi, 2006), where polyphonic lines of voice/cello and of saxophone/viola are separated from monoaural mixtures.

For example, Hainsworth et al. (Hainsworth and Macleod, 2001) proposed a method to get the genre using the bass line from an audio file. In the same way, Ryynanen (Ryynanen and Klapuri, 2007; Ryynänen and Klapuri, 2008) extracts multiple F0 frequencies from audio files to obtain the bass line using *Variable Markov Models* and bigrams to classify the genre for the audio file.

Other research lines aim to obtain information from other instruments. Paulus et al. (Paulus and Klapuri, 2009) used *Hidden Markov Models* to extract the drum line from audio files in order to get the rhythm features from the song. In a similar way, in (Tsunoo et al., 2009) the authors developed a method to classify genre based on rhythmic patterns. After extracting unit rhythmic patterns from a number of audio tracks they propose a pattern occurrence histogram for genre classification.

In the symbolic domain, this information is usually already separated in parts or voices. Usually a different instrument is assigned to each part, but sometimes they are a way to structure music material. Usually MIDI files contains a track with the melody, but it is possible that the MIDI file contains more than one melody track. This could be a problem for algorithms that get only one melody track, but only in the case that all the melodic information needs to be extracted. But if the method selects a part that is a valid melodic line, it could be enough for many applications.

Extracted features from the melody line are often used to index large databases of MIDI files or any other symbolic formats. For example, works like (Ponce de León and Iñesta, 2007) classify files in genres from the information contained in the melody track. Hence it is necessary to create automatic systems to recognise and extract melody and bass lines from symbolic files. Rizo et al. (Rizo et al., 2006a; Rizo et al., 2006b) have obtained promising results using a *Random Forest*-based algorithm to select and extract the melody track from MIDI files. In a similar way, Jiangtao et al. (Jiangtao et al., 2009) proposed the use of neural networks to get the melody track.

Ozcan et al. (Ozcan et al., 2005) approached a method for melody extraction based on the elimination of MIDI channels not containing melodic information using pitch histograms. Eventually the monophonic melody line is obtained using skyline algorithms (Uitdenbogerd and Zobel, 1998) over the remaining tracks. In a similar work Madsen et al. (Madsen et al., 2010) presented a reduction algorithm for MIDI files to preserve the relevant musical content in order to improve indexing and searching in musical databases.

Simsekli (Simsekli, 2010) showed the importance of using the bass line from MIDI files to classify these files by genre. So far, however, there has been little discussion about the bass line extraction from music pieces in the symbolic domain.

For this study, jazz, classical music and modern popular music MIDI files have been used. The datasets used are the same than those utilized in former works (Ponce de León, 2011) in order to compare our results with theirs. This comparison will permit us to evaluate comparatively the difficulty of both tasks.

Section 3 describes the design of the track descriptors and the classifiers. Section 4 presents an overview of the dataset utilized. Section 5 deals with the implementation of the dictionary for semi-automatic track labeling. Section 6 presents the results of the experimental set-up, and finally, in Section 7 conclusions and future perspectives are drawn.

The generic scheme for how the data are managed in this work is shown in Figure 1. From the MIDI file repository all the track labels are extracted. Those related with ‘bass’ parts are selected for building up a bass dictionary. With the help of this dictionary, a set

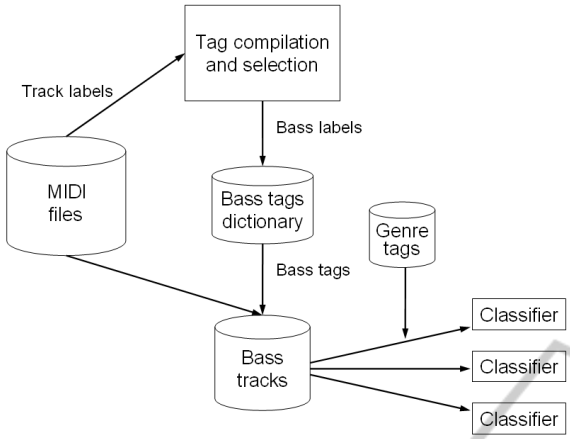

Figure 1: Generic scheme for dictionary and classifiers implementation.

of bass tracks are selected that will be used to train a number of classifiers for different music genres in order to check the specificities of bass parts for different genres.

3 MIDI TRACK DESCRIPTION AND CLASSIFICATION

3.1 Statistical Descriptors

The music content description has been done using symbolic features extracted from the MIDI tracks. These descriptors are shown in table 1. Note that some of these descriptors are considered in both raw and normalized versions.

The normalization of a descriptor for a MIDI track has been done according to the values of the same descriptor for all the tracks in the same MIDI file using the next equation:

$$normalized_i = \frac{value_i - min}{max - min} \quad (1)$$

where $value_i$ is the value for a descriptor of the MIDI track i, max is its maximum value for all the tracks[1] in that MIDI file, and min is the minimum value of them. If all the tracks have the same value, 1 is considered for the normalized version, but this situation is extremely unusual in practice.

It is non sense to use non normalized versions of some descriptors, like for example, *Number of Notes* or *Duration* of the track, because the longer a music piece is, the longer their tracks will be, and this has little to do with its bass nature, although the relation between the number of notes in the bass track and in the other ones could be relevant.

[1] Actually, percussion tracks have note been considered. More information in Section 4.

Table 1: Normalized and non normalized features used for track content description.

Category	Normalized	Non normalized
Track information	Avg Polyphony	Avg Polyphony
	Duration	-
	Occupation	-
	Occupation Rate	Occupation Rate
	Number of Notes	-
Pitch	Highest	Highest
	Lowest	Lowest
	Mean	Mean
	Standard Deviation	Standard Deviation
Pitch intervals	Largest	Largest
	Smallest	Smallest
	Mean	Mean
	Standard Deviation	Standard Deviation
Note durations	Longest	Longest
	Shortest	Shortest
	Mean	Mean
	Standard Deviation	Standard Deviation

Figures 2 and 3 show how some descriptors take different ranges of values for both classes. Figure 2 shows the non normalized descriptors *Avg Polyphony* and *Low Pitch*. It is possible to see that *Avg Polyphony* is always near a value 1 for *bass* tracks. In the same way, *bass* tracks often have lower values for the pitch values, as expected. Figure 3 shows similar behaviours for the normalized descriptors *Mean Pitch* and *Number of Notes*.

There are four groups of descriptors. *Track Information* descriptors are features that describe the whole track in relation to the others in the same MIDI file. *Avg Polyphony* determines the polyphony degree of a track, measured as the ratio between the number of notes in the *skyline* reduction of a track and the total number of notes in it. *Duration* and *Occupation* descriptors measure the duration from the first to the last note of a track and the time during which at least one note is sounding. *Occupation Rate* is the ratio between *Occupation* and *Duration*.

Pitch group provides information about the notes. The maximum possible pitch is 127 corresponding to the note G_8, and the minimum possible pitch is 0 for the note C_{-2}, although both extreme values are very unusual pitches, since piano pitches range from A_0 (21) to C_8 (108). The highest, lowest, mean pitch, and deviation are computed.

Pitch Interval group describes the information about the horizontal melodic intervals between consecutive notes. Intervals are considered in absolute values. These descriptors have been obtained from the *skyline* reduction for each MIDI track because

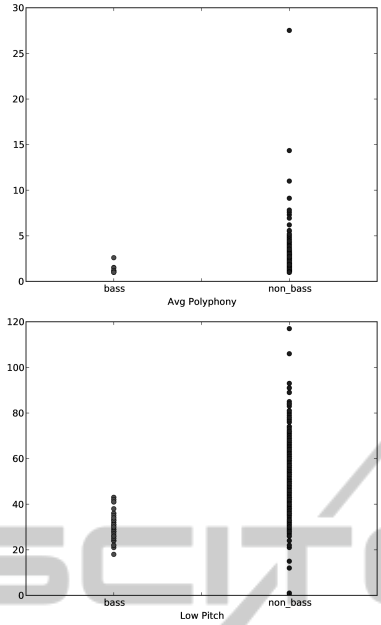

Figure 2: Examples of non-normalized symbolic features used to track description. Top: *Avg Polyphony* values for *bass* and *non-bass* tracks. Bottom: *Low Pitch* values for *bass* and *non-bass* tracks.

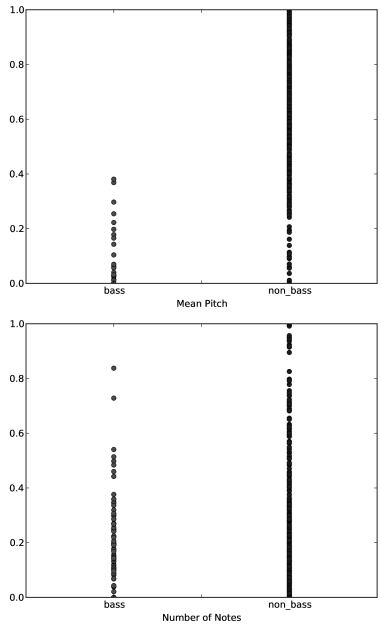

Figure 3: Examples of normalized symbolic descriptors. Top: the *Mean Pitch* descriptor for *bass* and *non-bass* tracks. Bottom: *Number of Notes*.

they are only applicable to monophonic sequences. Largest, smallest, mean interval and deviation are computed.

Note duration group provides the information about how long are the notes in a MIDI track. The duration unit used in MIDI files for this value is *beat*. For that, the durations, usually given in MIDI ticks, have been divided by the MIDI file resolution. Longest, shortest, mean and deviation are computed in this group.

3.2 Bass Track Classification

For the classifier implementation, the WEKA toolkit (Hall et al., 2009; Witten and Frank, 2005) has been used, utilizing the *Random Forest* (RF) algorithm (Breiman, 2001). It has been selected due to their ability for making their own feature selection in a natural way, exempting us from performing a, possibly method-conditioned, feature selection analysis.

RFs are able to assign a class probability to the samples. This way, a sample description vector for a MIDI track $\mathbf{t}$ is classified by each tree in the forest and their decisions are combined giving as a result a membership probability for the class.

Given a RF with K trees, where each tree T_j outputs decision d_j on an input sample $\mathbf{t}$, the probability of this track of being a bass track will be denoted as $p(B|\mathbf{t})$, and computed as

$$p(B|\mathbf{t}) = \frac{\sum_j w_j \delta(B, d_j)}{K} \tag{2}$$

where

$$\delta(B, d_j) = \begin{cases} 1 & \text{if } d_j \text{ is Bass} = \text{TRUE} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

and w_j is a weight factor for each T_j as a purity coefficient, computed as the ratio between the number of samples of the winning class for the leaf from which the decision is given (majority class) and the total number of samples represented by that leaf. Therefore, $w_j \in]0.5,1]$.

The probability $p(B|\mathbf{t})$ is useful to determine whether a track $\mathbf{t}$ is a bass line or not. This is a binary distribution, since $p(B|\mathbf{t}) + p(\bar{B}|\mathbf{t}) = 1$, so a given track will be considered as a bass track when $p(B|\mathbf{t}) > 0.5$. This task can be regarded as a two class classification problem, and the performance will be assessed through the number of correct decisions (accuracy) in percentage.

3.3 Bass Track Selection in Multi-track Files

The problem of deciding which track of a multi-track MIDI file contains the bass line is addressed in a different way. For that, a probability $p(i|B)$ must be computed, considering that $p(B|i) = p(B|\mathbf{t}_i)$, in the

context of the rest of tracks in the file $i = 1, 2, ..., N$. Applying Bayes,

$$p(i|B) = \frac{p(i)p(B|i)}{p(B)} = \frac{p(i)p(B|i)}{\sum_{j=1}^{N} p(j)p(B|j)} \quad (4)$$

but we are working under the hypothesis that the priors $p(i)$ are the same for all tracks, and equal the bass to non bass track amount ratio in the database, regardless of the particular file considered. Therefore, the equation above simplifies to just a normalization

$$p(i|B) = \frac{p(B|i)}{\sum_{j=1}^{N} p(B|j)} \quad (5)$$

Once this posterior probabilities are computed, the decision is taken under a maximum likelihood approach, so the selected track number will be

$$\hat{i_B} = \arg\max_i p(i|B) \quad (6)$$

A problem arises here if none of the tracks contains a bass line. In such a case $p(B|\mathbf{t}_i)$ should be very low for all i. To manage this possible scenario, we have considered a virtual zero track with $i = 0$ and a fixed probability $p(B|\mathbf{t}_0) = \theta$ has been assigned to it. This probability is introduced in the equation (5)

$$p(i|B) = \frac{p(B|i)}{\sum_{j=0}^{N} p(B|j)} \quad (7)$$

and equation (6) keeps working. If $\hat{i_B} = 0$ the system is saying that none of the tracks contains a bass line.

It seems sensible to establish $\theta = 0.5$. This way, no assumptions about its bass character are assumed, but in practice this value acts as a threshold parameter, in such a way that only those tracks with $p(B|i) > \theta$ are considered for being selected, so it can be tuned heuristically.

In order to know the goodness of the selection, the *Precision*, *Recall* and *F-measure* parameters have been used, defined as follows:

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

$$F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (10)$$

where FP is the number of false-positives (the classifier selects a non-bass track), TP is the number of true-positives (the selected track contains the correct bass line), and FN is the number of false-negatives (the classifier does not select any track but the MIDI file indeed contains at least one bass track).

The selection is considered as successful both if the selected track contains the correct bass line or the MIDI file has not any bass track and the classifier does not select any bass track (a true-negative situation).

Precision offers an indicator of the hit rate to select the bass tracks properly. A low value for *precision* indicates that, for a large number of files, *non bass* tracks have been selected as *bass*. *Recall* offers an indicator of the classifier ability to detect the bass tracks. A low value of *recall* indicates that, for a large number of files, the *bass* track has not been identified. *F-measure* is the harmonic mean of *precision* and *recall*, and it can be interpreted as a weighted average of them.

3.4 Improving Track Categorization using Multimodal Information

A question of multimodal nature can be posed at this point: can we make use of knowing which track contains the bass line to improve melody track detection? If bass line probabilities are well estimated they might be used for improving the quality of melody track selection. Bass track is selected using the same scheme described above, so

$$\hat{i_M} = \arg\max_i p(i|M) \quad (11)$$

with the probabilities $p(i|M)$ estimated in the same way from the corresponding $p(M|\mathbf{t}_i)$ computed by the random forests trained with melody data.

A constraint is set that melody and bass lines must be in different tracks. This way, $\hat{i_M} \neq \hat{i_B}$.

A first naïve approach could be to remove the selected bass track before making the melody line selection. This would simplify the problem (less tracks) but it does not make use of the new information available. For that, instead of looking at $p(i|M)$ for the remaining tracks, we could look at the probabilities of being a melody conditioned also by the knowledge of how a bass looks like and under the constraint that the bass line can not be in the same track as the melody. We will denote this probability as $p(i|M, \hat{i_B} \neq i)$.

For computing this probability we assume that bass and melody tracks are not mutually conditioned, and therefore we should look for the most likely combination for the i that contains the melody, while j contains the bass, $p(i|M)p(j|B)$, and then make the summation for all different combinations, for a given i. This is,

$$p(i|M, \hat{i_B}) = \sum_j p(i|M)p(j|B)\ . \quad (12)$$

The constraint $\hat{i_M} \neq \hat{i_B}$ implies that $p(i|M)p(i|B) = 0$, so

$$p(i|M, \hat{i_B} \neq i) = \sum_{j \neq i} p(i|M)p(j|B) \quad (13)$$

and, since $p(i|M)$ is the same in all cases,

$$p(i|M,\hat{i_B} \neq i) = p(i|M)\sum_{j\neq i} p(j|B) \quad (14)$$

$$= p(i|M)(1-p(i|B)) \,. \quad (15)$$

They should be divided by a constant $C = \sum_{k,l=0;k\neq l}^{N} p(k|M)p(l|B)$ for keeping normalization, but it can be skipped for classification purposes, since it is the same for all tracks. Thus, eventually, eq. (15) is what we can use for classification:

$$\hat{i_M} = \arg\max_i \{p(i|M)(1-p(i|B))\} \,. \quad (16)$$

Of course, this reasoning and therefore the equations are reversible in terms of the track categories, so we could also use the melody track information for helping the bass track selection through $p(i|B,\hat{i_M} \neq i)$. In fact, both approaches will be tested in the experiments section.

4 DATASETS

Three sets of MIDI files have been used for the classifier implementation: KR200, CL200, and JZ200. KR200 is a set of 200 MIDI files of modern popular music in the karaoke format (*.kar*)[2], CL200 is a set of 200 MIDI files of classical music, and JZ200 is a set of 200 MIDI files of jazz music. The bass track in these MIDI files have been tagged using the procedure described in the next section and manually checked for correction afterwards.

Instruments played through channel 10 in General MIDI files are percussion sounds and, therefore, usually they do not contain melodic features. For instance, pitch 38 in an instrument played by channel 10 is a snare drum. This instrument provides horizontal interval features as hits in rhythmic patterns, but it does not contain any pitch information. Due to this reason, tracks played through channel 10 have been removed for the database compilation.

These datasets are the same as those utilized in former researches (Rizo et al., 2006a; Ponce de León, 2011), where they were tagged according to the melody part selection task, so we have added the new bass tag to the existing melody tag. In any of the files, bass and melody were in the same track.

All the datasets are available to other researchers under request to the authors.

[2] A kind of standard MIDI file including the song lyrics in it, but fully compatible with the standard.

5 DICTIONARY-BASED TAGGING

In order to build the training sets correctly, the datasets have to be properly tagged as *bass* or *non-bass*. A first approach is to use the already present track names for tagging. This can not be a fully automatic procedure, since it is possible that a particular instrument or part can be named in different ways, and in addition, the names may not be correct. For instance, one bass track could be named as *bass*, or *electric bass*, *upright*, or *tuba*, for example. There is also a multilingual issue that should be taken into account (See the appendix for a list of all the tags found).

For approaching this problem, all the track names in the databases were collected and a dictionary was built with them. In order to avoid dispersion, these tags were modified converting uppercase characters to lowercase, and removing all white spaces. The amount of different tags related to bass and non bass tracks is shown in Table 2.

Table 2: Number of different tags related to bass and non bass content.

	Number of different tags
All tags obtained	732
Bass tags obtained	36

As an illustration, the tags for bass tracks that appeared at least four times are shown in table 3. The whole list can be found in the appendix.

Table 3: Most frequent bass tags.

Bass tag	Number of repetitions
fingeredbass	4
synthbass	4
bajo	8
basse2	16
basse	21
bass	94
bass-ok	125
bass(bb)	296

The way in which the tracks are labeled is a semi-automatic procedure. The tracks with a name containing one of these bass labels is tagged as a bass part automatically, but then a manual check has been performed to prevent errors.

It is very important, in order to analyze the results, to note that popular music and jazz usually have just one bass track, but it is possible to find files with more than one bass track, while other sequences may have any. Therefore, we will consider a successful decision the selection of any track from a file if it is tagged as

a bass part. For example, for classical music is easier to find several tracks to which the role of bass part can be assigned, mainly in orchestral pieces. Anyway, when we have performed a statistical analysis to our datasets, we have found that this is not the case in these particular databases (see Table 4), but this is an issue that must be taken into account for the experimental design.

Table 4: MIDI files classified by the number of bass tracks.

	KR200	CL200	JZ200
No bass tracks	6	13	0
One bass track	173	186	200
$>$ one bass tracks	21	1	0

In table 5, the number of tracks used for the classifier learning for all the datasets is shown.

Table 5: Number of *bass* and *non-bass* tracks in the MIDI datasets. Proportions of both kind of tracks per style are also shown.

Dataset	Bass tracks	Non bass tracks	Total
CL200	188 (27.3%)	500 (72.7%)	688
JZ200	200 (26.4%)	558 (73.6%)	758
KR200	221 (13.2%)	1456 (86.8%)	1677

6 EXPERIMENTS AND RESULTS

This section is structured in different parts, one for each of the experiments that were carried out. The first one is the classification of a given track as a bass or non bass line. The second is the selection of the bass track among those contained in given a multitrack standard MIDI file. This latter problem is also analyzed from the point of view of how much it is affected by the music genre of the data, and how sensitive is to training with a particular dataset. The last experiment tries to assess the multimodal approach to part selection.

As a reference work, the results obtained will be compared to those in (Ponce de León, 2011) for the melody track selection task.

6.1 Bass versus Non-bass Classification

This experiment consists of three sub-experiments. Each one implements a classifier for each of the genres of the datasets, CL200, JZ200 and KR200, using the RF algorithm and a 10-folded cross-validation scheme. Some preliminary trials were made, in which the RF classifier was configured using different values for both the number of trees in the forest, K, and that of the randomly selected features, F. The best results were obtained for $K = 10$ and $F = 6$, so this configuration will be utilized in all the experiments in the rest of the paper. Results obtained for this classification are shown in Table 6.

Table 6: Percentages of successful *bass* versus *non-bass* classifications and *melody* versus *non-melody* classification.

Dataset	Bass	Melody
CL200	98.7 ± 1.3	99.1 ± 0.7
JZ200	100	96.8 ± 1.4
KR200	97.2 ± 1.7	96.8 ± 1.8
All	98.6 ± 1.0	97.5 ± 1.3

The obtained results were very accurate, improving those obtained in melody classification. The best results were obtained for jazz, where a 100% of bass track classification was obtained, showing that bass line in the jazz database contained very characteristic features, and also that it is very uniformly constructed (see Table 4). Anyway, we are persuaded that this result is a very optimistic situation, and a poorer performance is expected in more varied, realistic datasets. For the other genres, results were comparable in average to those obtained for melody.

Accuracy classification for KR200 dataset is surprisingly lower than for the other genres. It seems that the KR200 corpus is harder for the bass classification. There are no a priori style-based difficulties for this lower precision, so it has to be caused by the way these MIDI files have been sequenced. In addition, a mix of very different sub-styles like rock, pop or hip-hop have been found in this dataset, what can difficult the learning of style specificities.

In any case, the precision obtained for the bass classification for the KR200 dataset has also improved the results for the melody classification, although the differences are not significant.

6.2 Bass Track Selection

Here, the target is a multi-track MIDI file, and the system outputs the number of the track selected as bass line, $\hat{i}_B$. If the system outputs $\hat{i}_B = 0$ means that none of the tracks was selected (see Section 3.3).

The number of MIDI files is lower than that of tracks, so leave-one-out has been used in this case.

A track i is considered as a candidate if its probability is higher than the threshold value θ as defined in section 3.3, therefore $P(B|i) > \theta$. The selected bass track is the one that maximizes $P(B|i)$. After some preliminary experiments, we have found that the best results were obtained for $\theta = 0.25$, so this value has been used for all experiments.

The obtained results for bass track selection are shown in Table 7, together with those previously obtained for melody track selection in a similar experiment.

Table 7: Bass track selection results and comparison with melody track selection.

	Dataset	Acc.%	Prec.	Rec.	F-m
Bass	CL200	95.0	0.95	1.00	0.97
track	JZ200	100	1.00	1.00	1.00
selection	KR200	96.5	0.97	0.99	0.98
Melody	CL200	100	1.00	1.00	1.00
track	JZ200	96.5	0.97	0.99	0.98
selection	KR200	72.3	0.72	0.99	0.84

Note that, again, the performance was very accurate. Specially, in the case of the KR200 corpus, where a 24% of improvement was obtained for the accuracy with respect to melody selection. A lower accuracy (-5%, due to lower precision, false positives in files without any bass track) was obtained for classical music, showing that the bass line is harder to characterize for this genre, as expected. Anyway, when only classical MIDIs containing at least one bass track were considered, a 100% of accuracy was obtained.

In general, these results suggest that bass track characterization is easier than melody. This is no surprise, since it seems that bass lines are a more concise concept than melodies, that are harder to define, even for expert musicologists.

6.3 Bass Track Selection across Styles

This experiment tries to determine the adaptability of the proposed method for different styles of music. Now the system is trained with the MIDI files in two of the datasets we have, and tested with those from the genre that was not used for training.

Three sub-experiments were made using the combination of two datasets (KR200 + JZ200 , CL200 + KR200, and CL200 + JZ200) for training and the other dataset for testing. The results obtained are shown in Table 8. Again, these results are compared to those previously obtained in a similar experiment of melody.

Note that these results are significantly better than those obtained for melody track selection for the same data and descriptors, but are poorer than those obtained when training and testing data were taken from the same music genre (Table 7). This points to a genre dependency of the task. For example, there are a number of classical files that contain pieces sequenced for piano, using two tracks: right hand and left hand. If we consider that the piano left hand is what usually includes the bass line in piano music, and we count the selection of piano left hand as correct hits, then the accuracy rises to a 82.8%.

Table 8: Bass track selection and melody track selection across styles. Training was made in each case with the other sets not considered for testing.

	Test set	Acc.%	Prec.	Rec.	F-m
Bass	CL200	77.5	0.92	0.83	0.87
track	JZ200	100	1.00	1.00	1.00
selection	KR200	91.0	0.93	0.98	0.95
Melody	CL200	71.7	0.73	0.92	0.81
track	JZ200	92.6	0.96	0.97	0.96
selection	KR200	64.9	0.77	0.78	0.78

In any case, while the loss of accuracy in melody track selection when learning with songs from different genres was of a -13.2% in average, now it has yielded only a -7.7%, so it seems that bass line is less sensitive to this specificity than melody.

6.4 Multimodal Track Selection

This multimodal approach tries to evaluate how the melody information can help to select the proper bass track and vice-versa, following the methodology described in Section 3.4.

First, we will see how melody probabilities interact with bass track estimation to perform bass track selection. The melody probabilities for all the tracks in each file have been obtained applying the RF algorithm to the tracks that were previously tagged as melody (Ponce de León, 2011). The results from this experiment are shown in Table 9.

Table 9: Bass track selection using melody information.

Set	Acc.%	Prec.	Recall	F-m
CL200	100	1.00	1.00	1.00
JZ200	100	1.00	1.00	1.00
KR200	96.4	0.97	0.99	0.98

The results for Jazz and Popular music were roughly the same than those displayed in Table 7, but for classical music a rise to a 100% was obtained. This result shows that bass track description is more informative than melody description, so little new information is provided by knowing the melodicity of the tracks and the performance is not improved by adding such information, except for classical music, where the concept of bass line in less clearly-defined.

Nevertheless, in the second experiment, addressed to know how bass information can help to select the proper melody track, the results were quite different (see Table 10).

Table 10: Melody track selection using bass information.

Dataset	Acc.(%)	Prec.	Recall	F-m
CL200	99.5	0.99	1.00	0.99
JZ200	99.0	0.99	0.99	0.99
KR200	83.0	0.89	0.93	0.91

In this case results obtained have improved those obtained in Table 7 for JZ200 and KR200 datasets. Specially, in the case of KR200, where results improved from 72.3% to 83.0%. These significant improvements show the ability of the bass track description to lead melody selection to a better performance.

7 CONCLUSIONS AND FUTURE WORK

This work aims to select the bass line from a MIDI file using pattern recognition and machine learning techniques. Though there is a number of published works aimed to identify the melody track in MIDI files, it was not possible for us to find similar works for bass track selection in MIDI files.

The method employed is similar to that used for melody track characterization and selection in former researches (Rizo et al., 2006a; Ponce de León, 2011), and the results obtained in the experiments have been compared with theirs, using the same MIDI datasets for different music genres.

As we expected, this methodology works better for bass track classification than for, melody. It seems that bass line specificity is higher than the vague and less clearly-defined concept of 'melody'. The results achieved were very accurate, reaching a 100% for classical and jazz datasets.

For identifying and selecting the bass track from multitrack MIDI files, the results again improved those obtained for melody track identification. The achieved accuracy for bass using MIDI datasets that contains modern popular music was much higher (+24.2%) than those obtained for melody track classification with them. Although it was worse for classical music, where instrumental lines are less defined than for modern music.

Training and testing with the same style of music seems to be important for the successful selection of the bass line. The results were a 7.7% lower in average when the classifiers were trained with tracks of other genres. This should be a clue for the relevance of the bass track to work in genre classification using symbolic information.

On the other hand, the use of information from other tracks helped to select the bass or the melody track. The experiments showed that using bass information improved the melody track selection, although the improvement was lower when melody was used to select bass tracks.

In any case, all the conclusions have to be managed carefully due to the limited size of the employed data sets. Therefore, it will be necessary to use larger corpora to know the actual improvement when using the information from the other tracks. That research is conditioned by the long and tedious work of tagging and checking the ground truth in hundreds of MIDIs.

A total of 32 symbolic descriptors have been used to build the classifier, but there is not information about the relevance of each for the result achieved. The study of symbolic descriptor selection for the classifier remains as a future work to test if there is room for improvement. This way, an exhaustive search of possible combinations of descriptors could be used, but this implies unacceptable time conditions. Alternative techniques could be used to obtain the best combination of descriptors.

ACKNOWLEDGEMENTS

This work was supported by the project DRIMS (TIN2009-14247-C02), the Consolider Ingenio 2010 research programme (MIPRCV, CSD2007-00018), and the PASCAL2 Network of Excellence (IST-2007-216886). The authors would like to thank Jose Oncina and Jorge Calera-Rubio for their help and advice.

REFERENCES

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Hainsworth, S. and Macleod, M. (2001). Automatic bass line transcription from polyphonic music. In *Proceedings of the 2001 International Computer Music Conference*, pages 431–434. Citeseer.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. (2009). The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

Jiangtao, L., Xiaohong, Y., and Qingcai, C. (2009). *MIDI melody extraction based on improved neural network.* IEEE.

Kim, M. and Choi, S. (2006). Monaural music source separation: Nonnegativity, sparseness, and shift-invariance. In *Independent Component Analysis and Blind Signal Separation*, volume 3889 of *Lecture Notes in Computer Science*, pages 617–624. Springer Berlin / Heidelberg.

Lidy, T., Rauber, A., Pertusa, A., and Iñesta, J. (2007). Improving genre classification by combination of audio

and symbolic descriptors using a transcription system. In *Proc. of the 8th Int. Conf. on Music Information Retrieval, ISMIR 2007*, pages 61–66, Vienna, Austria.

Madsen, S., Typke, R., and Widmer, G. (2010). Automatic Reduction of MIDI Files Preserving Relevant Musical Content. In *Adaptive Multimedia Retrieval: Identifying, Summarizing, and Recommending Image and Music: 6th International Workshop, AMR 2008, Berlin, Germany, June 26-27, 2008. Revised Selected Papers*, volume 5811, page 89. Springer.

Ozcan, G., Isikhan, C., and Alpkock, A. (2005). *Melody Extraction on MIDI Music Files*. IEEE Computer Society.

Paulus, J. and Klapuri, A. (2009). Drum Sound Detection in Polyphonic Music with Hidden Markov Models. *EURASIP Journal on Audio, Speech, and Music Processing*, 2009:1–9.

Ponce de León, P. and Iñesta, J. (2007). Pattern Recognition Approach for Music Style Identification Using Shallow Statistical Descriptors. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 37(2):248–257.

Ponce de León, P. J. (2011). *A statistical pattern recognition approach to symbolic music classification*. PhD thesis, Computer Science, Alicante, Spain.

Rizo, D., Ponce de León, P. J., Pérez-sancho, C., Pertusa, A., and Iñesta, J. M. (2006a). A pattern recognition approach for melody track selection in midi files.

Rizo, D., Ponce de León, P. J., Pertusa, A., and Iñesta, J. (2006b). Melodic track identification in MIDI files.

Ryynanen, M. and Klapuri, A. (2007). Automatic bass line transcription from streaming polyphonic audio. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–1437. IEEE.

Ryynänen, M. P. and Klapuri, A. P. (2008). Automatic Transcription of Melody, Bass Line, and Chords in Polyphonic Music. *Computer Music Journal*, 32(3):72–86.

Simsekli, U. (2010). Automatic Music Genre Classification Using Bass Lines. *2010 20th International Conference on Pattern Recognition*, pages 4137–4140.

Tsunoo, E., Tzanetakis, G., Ono, N., and Sagayama, S. (2009). Audio genre classification using percussive pattern clustering combined with timbral features. *2009 IEEE International Conference on Multimedia and Expo*, pages 382–385.

Uitdenbogerd, A. and Zobel, J. (1998). Manipulation of music for melody matching. In *Proceedings of the sixth ACM international conference on Multimedia*, pages 235–240. ACM.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann Pub.

APPENDIX

Table with all the bass track names compiled from the databases, and the number of repetitions of each:

Name	#	Name	#
bassstrings	1	gtrbass	1
accbass	1	elecbassfinger	1
b6bass	1	fretlessbass	1
bass-elec.	1	electricbass	2
fingbass	1	bass2	2
bassintro	1	acousticbass	3
fingeredebass	1	basseac2	3
acouticbass	1	bassguitar	3
elbassfinger	1	basspicksolo	3
elecbass	1	synthbass	4
contrabass	1	tuba	4
fretlesse.bass	1	fingeredbass	4
acbass	1	bajo	8
bass1	1	basse2	16
electricbassfinger	1	basse	21
bass-rickenbck	1	bass	94
basssuraccomp	1	bass-ok	125
bassgtr	1	bass(bb)	296