Melody Track Identification in Music Symbolic Files

David Rizo, Pedro J. Ponce de León, Antonio Pertusa, Carlos Pérez-Sancho, José M. Iñesta

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante, Spain
{drizo,pierre,pertusa,cperez,inesta}@dlsi.ua.es
http://grfia.dlsi.ua.es

Abstract

Standard MIDI files contain data that can be considered as a symbolic representation of music (a digital score), and most of them are structured as a number of tracks, one of them usually containing the melodic line of the piece, while the other tracks contain the accompaniment. The objective of this work is to identify the track containing the melody using statistical properties of the notes and pattern recognition techniques. Finding that track is very useful for a number of applications, like melody matching when searching in MIDI databases or motif extraction, among others. First, a set of descriptors from each track of the target file are extracted. These descriptors are the input to a random forest classifier that assigns the probability of being a melodic line to each track. The track with the highest probability is selected as the one containing the melodic line of that MIDI file. Promising results have been obtained testing a number of databases of different music styles.

Introduction

There are different file formats to represent a digital score. Some of them are proprietary and others are open standards, like MIDI[1] or MusicXML[2], that have been adopted by many sequencers and score processors as data interchange formats. As a result of that, thousands of digital scores can be found on the Internet in these formats. A standard MIDI file is a representation of music designed to make it sound through electronic instruments and it is usually structured as a number of tracks, one for each voice of the music piece. One of them usually contains its melodic line, specially in the case of modern popular music. The melodic line (also called melody voice) is the leading part in a composition with accompaniment. The goal of this work is to automatically find this melodic line track in a MIDI file using statistical properties of the notes and pattern recognition techniques. The proposed methodology can be applied to other symbolic music file formats, because the information utilized to take the decision is only based on how the notes are arranged in each voice of the digital score. Only the feature extraction front-end is needed to be changed for dealing with other formats.

The identification of the melodic track is very useful for a number of applications. For example, melody matching when searching in MIDI databases, both in symbolic format (Uitdenbogerd & Zobel 1999) and in audio format (Ghias *et al.* 1995) (in this case, this problem is ofter named 'query by humming', and the first stage is often an identification of the notes in the sound query). In all these cases, search queries are always a small part of the melody and it should be clearly identified in the database files to perform melodic comparisons. Other application can be motif extraction to build music thumbnails for music collection indexing.

The literature about melody voice identification is quite poor. In the digital sound domain, several papers aim to extract the melodic line from audio files (Berenzweig & Ellis 2001; Eggink & Brown 2004). In the symbolic domain, Ghias and co-workers (Ghias *et al.* 1995) built a system to process MIDI files extracting something similar to the melodic line using simple heuristics not described in their paper and discarding the MIDI percussion channel.

In (Uitdenbogerd & Zobel 1998), four algorithms were developed for detecting the melodic line in polyphonic MIDI files[3], assuming that a melodic line is a monophonic[4] sequence of notes. These algorithms are based mainly in the note pitches; for example, keeping at every time the note of highest pitch from those that sound at that time (skyline algorithm).

Other kind of works focus on how to split a polyphonic source into a number of monophonic sequences by partitioning it into a set of melodies (Marsden 1992) or selecting at most one note at every time step (Uitdenbogerd & Zobel 1999). In general, these works are called monophonic reduction techniques (Lemstrom & Tarhio 2000). Different approaches, like using voice information (when available), average pitch, and entropy measures have been proposed.

Other approach, related to motif extraction, focus on the development of techniques for identifying patterns as repeti-

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

[2] http://www.recordare.com

[3] In polyphonic music there can be several notes sounding simultaneously.

[4] In a monophonic line no more than one note can be sounding simultaneously.

tions that are able to capture the most representative notes of a music piece (Cambouropoulos 1998; Lartillot 2003; Meudic 2003).

Nevertheless, in this work the aim is not to extract a monophonic line from a polyphonic score, but to decide which of the tracks contains the main melody in a multitrack standard MIDI file. For this, we need to assume that the melody is indeed contained in a single track. This is often the case of popular music.

The features that characterize melody and accompaniment voices must be defined in order to be able to select the melodic track. There are some features in a melodic track that, at first sight, seem to suffice to identify it, like the presence of higher pitches (see (Uitdenbogerd & Zobel 1998)) or being monophonic (actually, a melody can be defined as a monophonic sequence). Unfortunately, any empirical analysis will show that these hypotheses do not hold in general, and more sophisticated criteria need to be devised in order to take accurate decisions.

To overcome these problems, a classifier able to learn in a supervised manner what a melodic track is, based on note distribution statistics, has been utilized. For that, a number of training sets based on different music styles have been constructed consisting in multitrack standard MIDI files with all the tracks labelled either as melody or not melody. Each track is analyzed and represented by a vector of features, as described in the methodology section.

Therefore, a set of descriptors are extracted from each track of a target midifile, and these descriptors are the input to a classifier that assigns a probability of being a melodic line to each track. The tracks with a probability under a given threshold are filtered out, and then the one with the highest probability is selected as the melodic line for that file.

Several experiments were performed with different pattern recognition algorithms and the random forest classifier (Breiman 2001) yielded the best results. The WEKA (Witten & Frank 1999) toolkit was chosen to implement the system.

The rest of the paper is organized as follows: first the methodology is described, both the way to identify a melody track and how to select one track for a song. Then, the experiments to test the method are presented, and the paper finishes with some conclusions.

Methodology

MIDI Track characterization

The content of each track is characterized by a vector of statistical descriptors based on descriptive statistics of note pitches and durations that summarize track content information. This kind of statistical description of musical content is sometimes referred to as *shallow structure description* (Pickens 2001; Ponce de León, Iñesta, & Pérez-Sancho 2004).

A set of descriptors have been defined, based on several categories of features that assess melodic and rhythmic properties of a music sequence, as well as track properties. The list of the descriptors utilized is presented in table 1. The left column indicates the property analyzed and the right one the kind of statistics describing the property.

Table 1: Extracted descriptors

Category	Descriptors
Track information	Relative duration
	Number of notes
	Occupation rate
	Polyphony rate
Pitch	Highest
	Lowest
	Mean
	Standard deviation
Pitch intervals	Number of different intv.
	Largest
	Smallest
	Mean
	Mode
	Standard deviation
Note durations	Longest
	Shortest
	Mean
	Standard deviation
Syncopation	Number of Syncopated notes

Four features were designed to describe the track as a whole and 15 to describe particular aspects of its content. For these 15 descriptors, absolute and normalized relative versions have been computed. These latter descriptors were calculated using the formula $(value_i - min)/(max - min)$, where $value_i$ is the descriptor to be normalized corresponding to the i-th track, and min and max are, respectively, the minimum and maximum value for this descriptor for all the tracks of the target midifile. This permits to know these properties in terms of proportions to the other tracks in the same file, using non-dimensional values. This way, a total of $4 + 15 \times 2 = 34$ descriptors were initially computed for each track.

The track overall descriptors are its relative duration (using the same scheme as above), number of notes, occupation rate (proportion of the track length occupied by notes), and the polyphony rate, defined as the ratio between the number of ticks in the track where two or more notes are active simultaneously and the track duration in ticks.

Pitch descriptors are measured using MIDI pitch values. Maximum possible MIDI pitch is 127 (note G_8) and minimum is 0 (note C_{-2}).

The interval descriptors summarize information about the difference in pitch between consecutive notes. Pitch interval values are either positive, negative, or zero. Absolute values have been computed instead.

Note duration descriptors were computed in terms of beats, and are, therefore, independent from the midifile resolution.

Feature selection

The descriptors listed above are a complete list of all the features that have been computed, but any pattern recognition system needs to explore which are those features that

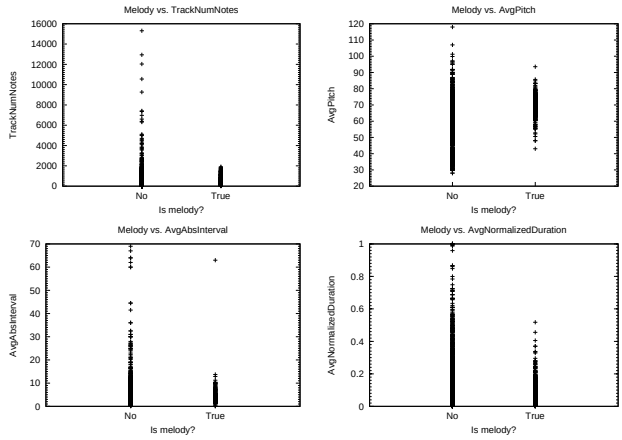

Figure 1: Distribution of values for some descriptors: (top-left) number of notes, (top-right) mean pitch, (bottom-left) mean absolute interval, and (bottom-right) mean relative duration.

actually are able to separate the target classes.

Some descriptors show evidence of statistically significant differences when comparing their distribution for melody and non-melody tracks, while other descriptors do not. This property is implicitly observed by the classification technique utilized (see below, in the "random forest" section), that performs a selection of features in order to take decisions.

A view to the graphs in Figure 1 provides some hints on how a melody track can look like. This way, a melody track seems to have less notes than other non-melody tracks, an average mean pitch, uses smaller intervals, and not too long notes. When this sort of hints are combined by the classifier, a decision is taken about the track "melodicity".

The random forest classifier

A number of classifiers were tested in an initial stage of this research and the random forests yielded the best results among them, so they were chosen for the experiments.

Random forests (Breiman 2001) are weighed combinations of tree classifiers that use a random selection of features to build the decision taken at each node. This classifier has shown good performance compared to other classifier ensembles with a high robustness with respect to noise. The forest consists of K trees. Each tree is built using CART (Duda, Hart, & Stork 2000) methodology to maximum size without pruning. The number F of randomly selected features to split on the training set at each node is fixed for all trees. After the trees have grown, new samples are classified by each tree and their results are combined, giving as a result a membership probability for each class.

In our case, the membership for class "melody" can be interpreted as the probability for a track of containing a melodic line.

Track selection procedure

There are MIDI files that contain more than one track suitable to be classified as melodic line: singing voice, instrumental solos, melodic introductions, etc. On the other hand, as usually happens in classical music, some songs do not have a well-defined melodic line, like in complex symphonies or single-track piano sequences. The algorithm proposed here can deal with the first case, but for the second, there are other methods like in (Uitdenbogerd & Zobel 1998) that perform melody extraction from polyphonic data.

In this work, only one track is selected as the melodic line and the method has to be able to deal with midifiles that may contain more than one melodic track. Therefore, given a file, all its tracks are classified and their probabilities p_i of being a melodic voice are obtained for $i = 1, 2, ..., N$ (the number of tracks in the file). Next, a threshold θ is applied in such a way that the condition $p_i \geq \theta$ is needed for track i to be considered as containing a melodic line. Thus, the method can detect no melody track in a file if $p_i < \theta, \forall i$.

Once the model has been able to discriminate between melodic tracks and non-melodic tracks, the problem of selecting just one as the melody line of the song is solved by picking the one having the highest probability.

Results

Data

Six corpora (see Table 2) were created, due to the lack of existing databases for this task. The files were downloaded from a number of freely accessible Internet sites. First, three corpora (JZ200, CL200, KR200) were created to set up the system and tune the parameter values. JZ200 contains jazz files, CL200 has classical music pieces where there was a clear melodic line, and KR200 containing popular music songs with a part to be sung (karaoke (.kar) format). All of them were composed of 200 files. Then 3 other corpora of the same music genres were compiled from a number of different sources to validate our method with different files and less homogeneous structure. This dataset is available for research purposes on request to the authors.

Corpus ID	Genre	No. of files	No. of tracks
CL200	Classical	200	687
JZ200	Jazz	200	769
KR200	Popular	200	1370
CLA	Classical	131	581
JAZ	Jazz	1023	4208
KAR	Popular	1360	9253

Table 2: Corpora utilized in the experiments, with identifier, music genre, number of files and total number of tracks.

The main difficulty for building the training sets was to label the melody track for each file. Instead of doing it manually, a meta-data based approach was used.

The number of midifiles downloaded for this work was originally of 27,956. For them, the track name meta-events were extracted, where available, and a statistical study was

performed. This way, we found that nine strings (lowercase) related to melodic content {*melody, melodie, melodia, vocal, chant, voice, lead voice, voix, lead, lead vocal, canto*}[5] appeared among the 50 most frequent track names included in the compiled data sets. A total of 24,806 different track names appeared.

A validation test is needed in order to assess the meta-data approach for labelling. This procedure conditions the ground truth used in the experimentation. MIDI meta-data are unreliable and are certainly not a guarantee, so a comparison to manual human expert labelling is needed, but it is unfeasible for the tens of thousands of tracks utilized. Thus, we have chosen 100 files randomly from those in the 6 corpora and compared the meta-data based labels to those manually assigned by a human expert. We have found a 98% of agreement between both decisions. Only two files appeared in which the names 'melody' and 'voice' were assigned to tracks that actually contained a melodic line but embedded in a polyphonic accompaniment track. In both cases, the midifiles had not a neat melody track. Anyway, this 2% of estimated error could be considered as an improvement margin for the results presented hereafter, if a human expert based labelling had been performed.

Therefore, only the files containing just one track named with one of the identifiers listed above were selected for the corpora. For each file, that track was labelled as melody and the remaining tracks were labelled as non-melody tracks. The percussion tracks (MIDI channel 10) and the empty tracks were not considered for the experiments. Summing all the corpora, 3,009 melody tracks and 13,859 non-melody tracks were utilized.

Threshold selection

An important issue for all the decisions is a proper selection of the threshold θ used for classification. This value must be selected in such a way that minimizes the error risk. The proposed procedure is to analyze the classifier's output after training for both melody and non-melody tracks when applied to the 2826 tracks from the 600 files contained in the JZ200, CL200, and KR200 corpora. From this output values in $[0, 1]$, a value for θ is assigned in order to minimize the number of classification errors.

The results from this test are displayed in Figure 2. Note the clear difference of the distributions for both types of tracks. With those data, the minimum error value is found for any $\theta \in [0.41, 0.59]$ (14 errors from 2826 tests). Thus, the value $\theta = 0.5$ in the middle of that interval was established and utilized for all the experiments hereafter.

Experiments

The WEKA package was used, and it was extended to compute the proposed track descriptors directly from MIDI files.

Four experiments were carried out. The first one tried to assess the capability of random forests to classify properly between melodic and non melodic tracks. In the second experiment, the aim was to evaluate how accurate is the system identifying the track that was actually the melody for each file. Finally, the specificity of system with respect to both the music genre and the corpora utilized were tested.

[5] for these strings alternative spellings have been considered, like the use of é, è or e; or the use of '1' after the name, like 'melody 1', 'lead 1' or 'voice 1'.

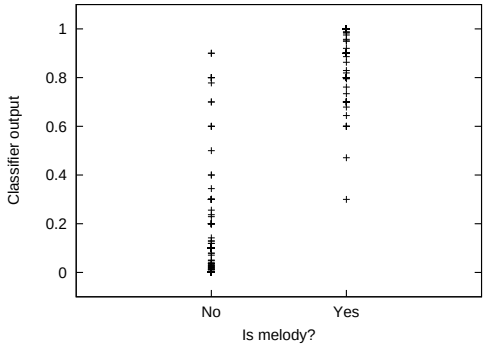

Figure 2: Distribution of classifier outputs for melody and non-melody tracks.

Corpus	Correctly classified tracks
CL200	99.2
JZ200	96.0
KR200	94.8

Table 3: Melody versus non-melody classification results (in percentages).

Melody versus non-melody classification Given a set of tracks, this experiment outputs the percentage of correctly classified ones. The three 200-file corpora (CL200, JZ200, and KR200) were used to test the system. This way, 2826 tracks provided by these files were classified in two classes: is melody / is not melody. A 10-folded cross-validation experiment was performed to estimate the accuracy of the method. The results are shown in Table 3. The excellent performance obtained is due to the fact that the classifier has been able to successfully map the input feature vector space to the class space. This can be also observed in Figure 2, where a clear separation of the classifier output was achieved for the samples of both classes. With the suitable selection of the value for the decision threshold ($\theta = 0.5$), only a few errors have been made.

This experiment has also been used to manually fix the values for the other parameters that have to be selected for the classifier. Thus, these results have been obtained using $K = 10$ trees and $F = 5$ features randomly selected for the random forest trees. This structure of the classifier is going to be used in the rest of experiments.

Melodic track selection Now, the goal is to know how many times the method has selected as melody track the proper track from a file. The input to the system is now the tracks in a file from which it selects the one with the highest melody probability value. To check the classifier's decision, the name metadata of the selected track has been verified to

be one of those included in the set of identifiers listed above.

The three 200-file corpora were used to test the system. Due to the more limited number of data available, this experiment was performed using a leave-one-out scheme to estimate the classification accuracy. The obtained results are shown in table 4.

Corpus	Correctly processed files
CL200	100.0
JZ200	99.0
KR200	86.6

Table 4: Melody track selection results. Percentages of files where the melody track was properly identified.

Note the high quality of the results. Now, a lower success rate has been obtained for the karaoke files due mainly to two factors: the presence in the training set of files with melody voices not properly tagged, according to the track identifier, so a number of melody tracks are labelled as not melody, confusing the training. And, conversely, missing tracks that are indeed the melody but having an identifier not included in the set of melodic identifiers.

Style specificity This experiment was designed in order to evaluate the system robustness against different corpora. In other words, how specific were the classifier's inferred rules with respect to the music genre of the files considered for training. For it, two experiments were performed: first, the classifier was trained and tested with files of the same genre (using both corpora for a given genre) (see Table 5), and secondly, the classifier was trained using the data of two styles and then tested with the files of the remaining style (see Table 6).

Style	% Success
Classical	70.0
Jazz	95.7
Karaoke	71.8

Table 5: Melody track selection within style. Percentages are on correctly processed files.

The results in Table 5 show that when more heterogeneous corpora are used, the performance lowers with respect to the 200-files corpora, that were collected from the same source, therefore sharing common structural patterns.

The lower performance for classical music is due to the difficulty in some classical styles for a particular track to be selected as the melody (e.g. canon pieces), where that role is continuously changing between tracks. In addition, we have verified the presence of very short sequences for the classical genre, causing less quality in the statistics that also influences a poorer result.

The results in Table 6 show that the performance is, in general, poorer (with respect to values in Table 5) when no data from the style tested were used for training. This is not true for classical music, due to effects related to the problems expressed above.

Training corpora	Test corpus	% Success
KAR+JAZ	CLA	71.7
CLA+KAR	JAZ	90.7
CLA+JAZ	KAR	52.2

Table 6: Melody track selection across styles. Percentages are on correctly processed files.

Training set specificity To see how conditioned are these results by the particular training sets utilized, a generalization study was carried out building a new training set with the three 200-files corpora, and then using the other corpora for test. The results are detailed in Table 7.

Training corpora	Test corpus	% Success
CL200+JZ200+KR200	CLA	76.3
CL200+JZ200+KR200	JAZ	95.6
CL200+JZ200+KR200	KAR	79.9

Table 7: Melody track selection by styles when training with data from all the styles.

When combining all the results, taking into account the different cardinalities of the test sets, the average successful melody track identification percentage was 86.1 %.

Conclusions and future work

A method to identify the voice containing the melodic line in a multitrack digital score has been proposed. It has been applied to standard MIDI files in which each line is structured in a different track, so the system determines whether a track is a melodic line or not. The one with the highest probability among the melodic tracks is finally labeled as the track containing the melody of that song.

The decisions are taken by a pattern recognition algorithm based on statistical descriptors of pitches, intervals, durations and lengths, extracted from each track of the target file. The classifier that performed the best among a number of them available in the WEKA toolkit was a kind of decision tree classifier named random forest. It was trained using MIDI tracks with the melody track previously labeled using a meta-data approach.

This system can be used for melody matching when searching in MIDI databases, because search queries are parts of the melody, or motif extraction to build music thumbnails for music collection indexing.

The experiments yielded promising results using databases from different music styles, like jazz, classical, and popular. Unfortunately, the results could not be compared to other systems because of the lack of similar works.

The results show that enough training data of each style are needed in order to successfully characterize the melody track, due to the specificities of melody and accompaniment in each style. Symphonic music is particularly hard for this task, because of the lack of a single track that corresponds to the melodic line. Instead, the melody is constantly changing

among different tracks along the piece. To overcome this problem, more sophisticated schemes oriented to melodic segmentation are needed, that are under research now.

Acknowledgements

This work was supported by the projects: Spanish CICyT TIC2003–08496–C04, partially supported by EU ERDF, and Generalitat Valenciana GV043–541. The authors would like to thank the reviewers for their valuable comments.

References

Berenzweig, A., and Ellis, D. 2001. Locating singing voice segments within music signals. In *Proceedings of the IEEE workshop on Applications on Signal Processing to Audio and Acoustics (WASPAA)*.

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

Cambouropoulos, E. 1998. *Towards a general computational theory of musical structure*. Ph.D. Dissertation, Faculty of music and Department of Artificial Intelligence, University of Edinburgh.

Duda, R. O.; Hart, P. E.; and Stork, D. G. 2000. *Pattern Classification*. John Wiley and Sons.

Eggink, J., and Brown, G. J. 2004. Extracting melody lines from complex audio. In *5th International Conference on Music Information Retrieval (ISMIR)*, 84–91.

Ghias, A.; Logan, J.; Chamberlin, D.; and Smith, B. C. 1995. Query by humming: Musical information retrieval in an audio database. In *Proc. of 3rd ACM Int. Conf. Multimedia*, 231–236.

Lartillot, O. 2003. Perception-based advanced description of abstract musical content. In Izquierdo, E., ed., *Digital Media Processing for Multimedia Interactive Services*, 320–323.

Lemstrom, K., and Tarhio, J. 2000. Searching monophonic patterns within polyphonic sources. In *Proceedings of the RIAO Conference, volume 2*, 1261–1278.

Marsden, A. 1992. Modelling the perception of musical voices: a case study in rule-based systems. In *Computer Representations and Models in Music*, 239–263. Academic Press.

Meudic, B. 2003. Automatic pattern extraction from polyphonic MIDI files. In *Proc. of Computer Music Modeling and Retrieval Conf.*, 124–142.

Pickens, J. 2001. A survey of feature selection techniques for music information retrieval. Technical report, Center for Intelligent Information Retrieval, Departament of Computer Science, University of Massachussetts.

Ponce de León, P. J.; Iñesta, J. M.; and Pérez-Sancho, C. 2004. A shallow description framework for musical style recognition. *Lecture Notes in Computer Science* 3138:871–879.

Uitdenbogerd, A. L., and Zobel, J. 1998. Manipulation of music for melody matching. In *Proceedings of the sixth ACM International Multimedia Conference*, 235–240. ACM Press.

Uitdenbogerd, A., and Zobel, J. 1999. Melodic matching techniques for large music databases. In *Proceedings of the seventh ACM International Multimedia Conference (Part 1)*, 57–66. ACM Press.

Witten, I. H., and Frank, E. 1999. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann.