AUTOMATIC CLASSIFICATION OF MIDI TRACKS

Alexandre Bernardo and Thibault Langlois
Universidade de Lisboa, Faculdade de Ciências, Departamento de Informática, Portugal

Keywords: MIDI Track Classification, Music Information Retrieval, Neural Networks, *k*-Nearest Neighbors.

Abstract: This paper presents a system for classifying MIDI tracks according to six predefined classes: Solo, Melody, Melody+Solo, Drums, Bass and Harmony. No metadata present in the MIDI file is used. The MIDI data (pitch of notes, onset time and note durations) are preprocessed in order to extract a set of features. These data sets are then used with several classifiers (Neural Networks, *k*-NN).

1 INTRODUCTION

Music Information Retrieval is, nowadays, a highly active branch of research and development in the computer science field, and focuses on several topics such as beat tracking, music genre classification, melody extraction, score-following, to name a few.

There are a lot of known applications that use this technology for some extent: the new generation media players, which organizes music in an intelligent way, based in the music itself, and generates, for example, dynamic playlists; Internet radio stations, which builds a playlist based on the user's taste; score following; finding similarities between songs in a large database.

The work, presented in this paper, focuses on music stored using MIDI format. Electronic instruments use this format to communicate and synchronize themselves. The format consists in a number of tracks where each track represents the sequence of notes (pitch level and duration) played by one instrument. MIDI files also contain some metadata, such as the instrumentation or key. One of the advantages of the MIDI format is its compactness. Many musical resources using this format are freely available on the Internet.

Previous work in Music Information Retrieval using MIDI format includes music genre detection where several approaches have been proposed. Some researchers use similarity measures based on Kolmogorov complexity estimates in conjunction with a classical Machine Learning technique like *k*-Nearest Neighbors (Ruppin and Yeshurun, 2006), Support Vector Machines (Li and Sleep, 2004b) or clustering (Cilbrasi et al., 2004). Cataltepe (Cataltepe et al., 2007) compares the performance of obtained with the Normalized Compression Distance approach on MIDI and audio files with the ad-hoc features extraction and Machine Learning approach proposed by McKay (McKay and Fujinaga, 2004).

Other researchers proposed to extract a set of features from MIDI files and perform a genre classification using Neural Networks (McKay and Fujinaga, 2004) (Huang et al., 2004) or Support Vector Machines in conjunction with dimensionality reduction techniques (Li and Sleep, 2004a). Basili (Basili et al., 2004) made a comparison of various Machine Learning techniques on a musical genre classification task.

Another approach is to perform automatic melody detection. Rizo et al. (D. et al., 2006a) (D. et al., 2006b) has proposed a set of features to characterize each MIDI track and used a Random Forest classifier to identify tracks which contain melody. In (Madsen and Widmer, 2007), an information-theoretic complexity measure and an estimate of the local entropy are used to recognize melody tracks.

In this paper we address the problem of MIDI track classification. Based on the pitch levels and durations which describe each track, we extract a set of features that are used to train a classifier. It is important to note that, in contrast to many other previously published studies, our approach does not use any metadata present in the MIDI file (such as instrumentation). Tracks are classified into six classes: Solo, Melody, Melody+Solo, Drums, Bass and Harmony. Two Machine Learning approaches are compared. The rest of the paper is organized in the following way: section 2 describes the data and the different sets of descriptors and the classifiers that were used. Section 3 reports the experiments and the results obtained. Finally, section 4 concludes and discusses future directions of research.

Bernardo A. and Langlois T. (2008).
AUTOMATIC CLASSIFICATION OF MIDI TRACKS.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - AIDSS*, pages 539-543
DOI: 10.5220/0001708705390543
Copyright © SciTePress

2 METHODOLOGY

In order to characterize the musical content from each track, a vector of numeric descriptors, normally known as shallow structure description, is extracted. Then they are used as inputs for the classifiers — Neural Network and *k*-Nearest Neighbors — which were implemented in the Matlab environment. Also, the MidiToolbox Matlab toolbox was used for handling the MIDI files.

2.1 MIDI Track Description

Each MIDI track is characterized by a vector of numeric descriptors, such as pitch, note and silences information, which summarizes the track musical content and provides a statistical overview of the track. Based on other similar works in this area, twenty five descriptors, plus twelve more that represent the pitch intervals histogram, have been defined, and are presented in Table 1. There are five descriptors for track information, used to represent the track as a whole, and thirty two other descriptors for specific characteristics, which are subdivided into seven categories. Normalized values are computed for all descriptors, except the Intervals Histogram, so there's a proportional relation between all tracks from the same MIDI file.

The first category, Track Information, has five descriptors: duration, the track duration in beats; number of notes; number of significant silences, which are silences greater than a tick (1/16 beat) - smaller silences are not considered silences, as they acknowledgments are almost imperceptible and non-significant; occupation rate, which is the proportion of the track occupied by notes; polyphony rate, a proportion of the track occupied by two or more simultaneous notes. Pitch descriptors refer to the actual MIDI note value, ranging from 0 (C-2) to 127 (G8). Pitch interval is the difference between two consecutive notes, and gives important feedback about the track melody/harmony progression, namely, in respect to the number of different two-note intervals. *n*-grams descriptor also reflects the number of different pitch intervals, but on a three or four (depending on the track meter) consecutive notes basis. Note durations descriptors are self explanatory, as Silences durations, and are computed in beats. Syncopation is a rhythmic descriptor which reflects the number of notes whose onset is after the beat, normally in between beats. Syncopation is very frequent in jazz, and is also an important aspect to consider. Pitch intervals histograms show the frequency of the intervals semitones, giving valuable information about the musical scale and the kind of melody, or harmony, of the track.

2.2 Classifiers

Two different classifiers have been used to train and test the system, a Neural Network (NN), which is the main classifier, and *k*-Nearest Neighbors (*k*NN) for comparison purposes and for validating some decision choices on which descriptors should be used for best results.

2.2.1 Neural Network

Several Neural Networks (Multi-layer Perceptrons) were created for these experiments with a number of hidden units ranging from 40 to 100, in the search for the best balance between hidden units/computing time/results. Multi-Layer Perceptrons were trained using the scaled conjugate gradient algorithm.

2.2.2 K Nearest-Neighbors

k-Nearest Neighbors approach provides very fast results, given limited data files, and gives us the capacity to steer the experiments in the right direction.

Table 1: Descriptors.

Category		Descriptors
Track Information (TI)	1	Duration
	2	# Notes
	3	# Significant silences
	4	Occupation rate
	5	Polyphony rate
Pitch (P)	6	Highest
	7	Lowest
	8	Mean
	9	Standard Deviation
Pitch Intervals (PI)	10	# Different intervals
	11	Largest
	12	Smallest
	13	Mean
	14	Mode
	15	Standard Deviation
Note Durations (ND)	16	Longest
	17	Shortest
	18	Mean
	19	Standard Deviation
Silences Durations (SD)	20	Longest
	21	Shortest
	22	Mean
	23	Standard Deviation
Syncopation (S)	24	# Syncopated notes
Repetitions (R)	25	# Different n-grams
Intervals Histogram (semitones) (IH)	26	(0)Perfect Unison
	27	(1)Minor Second
	28	(2)Major Second
	29	(3)Minor Third
	30	(4)Major Third
	31	(5)Perfect Fourth
	32	(6)Augmented Fourth, Diminished Fifth
	33	(7)Perfect Fifth
	34	(8)Minor Sixth
	35	(9)Major Sixth
	36	(10)Minor Seventh
	37	(11)Major Seventh

2.3 MIDI File Format

The MIDI format has several ways to organize each track, and unfortunately, there is no real standard, because there are numerous ways, MIDI sequencers, to create a MIDI song, and each MIDI sequencer may create the MIDI file in a different way. This can lead to several problems when interpreting the MIDI (i.e. all the instruments on the same MIDI track but on different MIDI channels). Rosegarden, an open-source MIDI sequencer, was used to normalize all the MIDI files used in the experiments, so that all share the same structure.

2.4 Track Selection

A melody track can be interpreted as the leading voice, an instrument solo or simply a monophonic instrument playing its part throughout the song. The melody which we are interested in, is the leading voice. In a Jazz song there isn't always an obvious melody, but instead, several solos, or a melody and a solo on the same track. Harmony is, normally, provided by instruments such as piano, organ, guitar, or a suite of strings, and are polyphonic, which contrasts with melody or solo tracks, which are mostly monophonic. Harmony tracks may also contain solos, but these are mostly played in accompaniment with chords - a pianist soloing with the right hand, accompanies himself with the left hand - so it's harmony nevertheless. Also, bass and drums are categorized, mostly because they are evidently different from the other instruments, and are, individually and together, very important components in genre definition. We present six classes of tracks: Melody; Melody+Solo; Solo; Harmony; Bass; Drums. Tracks which don't fit in these classes are discarded.

2.5 Music Corpora

Two MIDI Music Corpora were assembled for the experiments, as depicted in Table 2, from only one genre, Jazz, as it incorporates the most common problems in identifying the different components of a song: jazz hasn't obvious singing voice melodies, has various solos on several tracks and most songs have enough instruments to populate our six classes with different data. This gives us enough different issues to solve in the descriptor extraction and classification methods. As the name implies, the neural networks were trained using the "training" set, and tests using the "test" set. [1]

[1]This music corpora is available for download at http://www.di.fc.ul.pt/tl/ICEIS2008/

Table 2: Music Corpora.

Corpus	Jazz (training)	Jazz (test)
# Files	40	43
# Tracks	239	252
Melody	37	41
Melody + Solo	23	18
Solo	23	22
Harmony	62	71
Bass	39	43
Drums	39	43

3 EXPERIMENTS

Early experiments showed that using all descriptors gives poor results, leading us to experiment with one descriptor category at a time, such as Pitch or Notes, or a combination of two categories. This approach led us to another problem. Which category combination was better? And which single descriptor combination? Testing every possible combination couldn't even be an hypothesis, for a very large number of combinations can be made out of all the thirty seven single descriptors.

A very simple algorithm solved the problem. The set of descriptors is built by testing each descriptor individually and joining iteratively more descriptors to the set while the performance increases. Section 3.3 shows some results obtained with different sets of descriptors.

The Matlab environment was used to implement the system and to perform the experiments. An additional toolbox was used, the MidiToolbox for helping with the handling of MIDI files in the Matlab environment. Matlab was chosen because it already sports a vast array of functions, classifiers, graphics, plots, etc, which help in analyzing the MIDI files.

3.1 Track Selection: All Descriptors

In the first set of experiments, all thirty seven descriptors were used, which proved to be a naive approach. Several NN were used, with hidden units ranging from 40 to 80, and they all presented basically the same results, varying only in 3%, so the best network was used, with 80 hidden units. *k*NN best *k* value was 7. The confusion matrices obtained with both methods are shown in tables 3 and 4.

Table 3: Confusion Matrix. NN: all descriptors. Classification Rate: 67,8%.

Melody	9	2	6	17	7	0
Melody+Solo	0	3	6	8	1	0
Solo	0	3	15	4	0	0
Harmony	0	0	5	61	3	2
Bass	0	0	0	1	42	0
Drums	0	0	0	1	1	41

Table 4: Confusion Matrix. *k*-NN: all descriptors. Classification Rate: 71,8%.

Melody	23	5	3	8	2	0
Melody+Solo	2	6	5	5	0	0
Solo	2	5	15	0	0	0
Harmony	5	3	1	57	3	2
Bass	1	1	0	2	39	0
Drums	0	0	1	1	0	41

k-NN gave slightly better results than NN, but not a significant benefit. Using all descriptors, is a naive approach, because some descriptors may successfully distinguish between two different classes, but another descriptor may distinguish the same classes in an opposite way, and confuse the final classification.

Notice the high score in Harmony, Bass and Drums, this is mostly because these tracks are quite different from each other, and specially from the other three classes. Harmony is polyphonic, as are mostly Drum tracks, in contrast to melody or solo tracks which are monophonic. Bass is also monophonic but usually has a lower pitch than melodies or solos, which makes it easy, for the classifier, to distinguish. It seems that the real problem is classifying Melody and Solo, as these are quite similar, and may be confused with Melody+Solo class.

3.2 Track Selection: Single Descriptor Category

A different approach was used in the following experiments. Instead of using the full set of descriptors, six sets of descriptors were used, corresponding to the descriptor categories, and also some combinations of the best scoring sets. Both networks used, with hidden units set to 80 for NN, and *k* values ranging from 1 to 29 for *k*NN, using the best value achieved. The results are shown in table 5.

With NN, surprisingly, the TI set alone provided better results than the whole set of descriptors. Also, the TI+P set provided the best results so far! All the other sets yielded worse results and are clearly confusing the classifier, and should not be used, at least not in this naive way. The *k*NN results, using set TI, were worse than those achieved when using the full set of descriptors, but using TI and P combined gave similar results. As in NN, all the other categories gave worse results. These results proved that the descriptors have to be carefully selected, not only by combining categories, but combining single descriptors, in order to achieve the best results.

Table 5: Single Categories Classification Rates.

Set	NN Rate(%)	*k*-NN Rate(%)
TI	71,8	63,8
P	56,7	53,9
PI	50,4	44,4
ND	50,4	42
SD	31,3	30,9
S	38,8	37,6
R	42	40,5
IH	47,2	36,9
TI+P	75,4	71,8
P+PI	63	48,8

3.3 Track Selection: Best Descriptors

Using the algorithm described previously a possible best descriptor set was found. The best set is composed of descriptors [1 2 4 5 7 8 9 15 18 24 31 34] (numbers are correspondent to table 1) for NN using 60 hidden units, and [3 4 5 6 7 8 9 11 13 18 19 24 25 31 33] for *k*NN with $k = 5$. It's clearly obvious that using a whole category is not the best option. Instead, using only the descriptors that work better together. For NN, which is the main classifier, a significant 16% gain was achieved comparing with the full descriptor set.

As we can see, only the descriptor #3 was not chosen from the TI set, which makes sense, as it was the set that provided the best results alone. From the P set, Highest Pitch was not chosen, but Lowest Pitch was, as it's used for classifying the bass tracks. From the PI set, only the Standard Deviation was used and from the ND set, only Mean was chosen. The SD set was ignored by the algorithm, which means that the silences are not significant and could be discarded. S was also chosen, meaning that the rhythm is also an important feature in distinguishing between classes.

The descriptors chosen from the IH set, were "Perfect Fourth" and "Minor Sixth". According to music theory, there intervals are one of the most consonant, because they have simple pitch relationships resulting in a high degree of consonance, which is perfect for distinguishing between, for example, a simple slow Melody or a fast complicated Solo.

In respect to the confusion matrix, all the misclassified tracks make sense. A melody track is similar to a bass track, although it has higher pitches. In fact, that one misclassified bass track has higher pitches as well. The Melody+Solo class is the worst performing, mainly because a solo can be made of several melodies, or even harmony at some point, and be misclassified. More training data would definitely improve the performance on this class.

Table 6: Confusion Matrix. NN: best descriptors. Classification Rate: 83,7%.

Melody	31	4	2	4	0	0
Melody+Solo	2	13	1	1	1	0
Solo	0	4	17	1	0	0
Harmony	2	0	0	69	0	0
Bass	0	1	0	0	42	0
Drums	0	0	2	1	1	39

Table 7: Confusion Matrix. KNN: best descriptors. Classification Rate: 77,3%.

Melody	28	4	3	4	2	0
Melody+Solo	2	10	5	1	0	0
Solo	4	2	14	1	1	0
Harmony	5	1	3	60	1	1
Bass	0	0	0	0	43	0
Drums	0	0	1	1	1	40

4 CONCLUSIONS AND FUTURE WORK

An Automatic Classification of Midi Tracks system has been implemented and presented. It uses MIDI files from a single genre, Jazz, and classifies the tracks in six classes, Melody, Melody+Solo, Solo, Harmony, Bass and Drums. A neural network is used to process thirty seven descriptors extracted from each MIDI track, which has been previously tagged in the six classes. The experiments showed that using all descriptors is a wrong approach, as there are descriptors which confuse the classifier. Using carefully selected descriptors proved to be the best way to classify these MIDI tracks. Future work, under research now, includes using a larger MIDI database, testing new genres, such as Rock, Pop or Classical, to prove the systems reliability between all genres, so that it can be used as a crucial part of a larger musical genre classification system. Having the tracks identified, as we have presented here, allows different processing specialized to each class.

ACKNOWLEDGEMENTS

This work was supported by EU and FCT, through LaSIGE Multiannual Funding Programme.

REFERENCES

Basili, R., Serafini, A., and Stellato, A. (2004). Classification of musical genre: a machine learning approach. In *ISMIR*.

Cataltepe, Z., Yaslan, Y., and Sonmez, A. (2007). Music genre classifcation using midi and audio featres. *Journal on Advances in Signal Processing*, 2007.

Cilbrasi, R., Vitányi, P., and de Wolf, R. (2004). Algorithmic clustering of music based on string compression. *Computer Music Journal*, 29(4):49–67.

D., R., de León P. J., P., C., P.-S., and Pertusa A., I. J. M. (2006a). A pattern recognition approach for melody track selection in midi files. In Dannenberg R., Lemström K., T. A., editor, *Proc. of the 7th Int. Symp. on Music Information Retrieval ISMIR 2006*, pages 61–66, Victoria, Canada. ISBN: 1-55058-349-2.

D., R., de León P.J., P., and Pertusa A., I. J. (2006b). Melodic track identification in midi files. In *Proc. of the 19th Int. FLAIRS Conference*. AAAI Press. ISBN: 978-1-57735-261-7.

Huang, Y.-P., Guo, G.-L., and Lu, C.-T. (2004). Using back propagation model to design a midi music classification system. In *International Computer Symposium*, pages 253–258, Taipei, Taiwan.

Li, M. and Sleep, R. (2004a). Improving melody classification by discriminant feature extraction and fusion. In *Proc. of the 5th Int. Symp. on Music Information Retrieval ISMIR 2004*.

Li, M. and Sleep, R. (2004b). Melody classification using a similarity metric based on kolmogorov complexity. In *Sound and Music Computing*, Paris, France.

Madsen, S. T. and Widmer, G. (2007). A complexity-based approach to melody track identification in midi files. In *International Workshop on Artificial Intelligence and Music (MUSIC-AI 2007)*, Hyderabad, India.

McKay, C. and Fujinaga, I. (2004). Automatic genre classification using large high-level musical feature sets. In *ISMIR*.

Ruppin, A. and Yeshurun, H. (2006). Midi music genre classification by invariant features. In *ISMIR*, pages 397–399.