A Complexity-based Approach to Melody Track Identification in MIDI Files

Søren Tjagvad Madsen[1] and Gerhard Widmer[2]

[1] Austrian Research Institute for Artificial Intelligence, Vienna
soren.madsen@ofai.at
[2] Department for Computational Perception, Johannes Kepler University, Linz
gerhard.widmer@jku.at

Abstract. In this paper, we will test the importance of music complexity as a factor for melody recognition in multi voiced popular music. The assumption is that the melody (or lead instrument) will contain the largest amount of information – will be less redundant than the rest. Measures of melodic complexity of pitch and timing are proposed. We test the different complexity measures and different prediction strategies, and evaluate them on the task of predicting which track of a MIDI file that contains the main melody. Filtering out melody tracks can be useful when searching large databases for similar songs. 108 melody track annotated pop songs were included in the experiment.

1 Introduction

Locating the melody in music is a trivial listening task. Human listeners are very effective in (unconsiously) picking out those notes in a – possibly complex – multi-voiced piece that constitute the melodic line. Melody is also an important aspect in music-related computater applications, for instance, in Music Information Retrieval (e.g., in music databases that offer retrieval by melodic motifs [1] or Query by Humming [2]).

Standard MIDI files are structured into tracks, and good sources have different instruments stored on different tracks. Current work of ours describes a method for locating the notes constituting a likely melody throughout a piece of classical music stored in a MIDI file [3]. The melody is constructed from notes from different instruments of the piece.

In this paper we assume that the melody role will be taken by a single instrument throughout the piece. This assumption is expected to hold in popular music. We address the problem of finding the single track of a MIDI file that holds the main melody of a pop song. Melody track identification is useful in systems that indend to change aspects of the melody in a MIDI file, e.g. changing the instrument or muting the melody in order to create a file suitable for karaoke.

2 Complexity and Melody Perception

The basic motivation for our model of melody track identification is the observation, which has been made many times in the literature on music cognition, that there seems to be a connection between the complexity of a musical line, and the amount of attention that will be devoted to it on the part of a listener. A voice introducing new or surprising musical material will potentially attract the listener's attention. However, if the new material is constantly repeated, we will pay less and less attention to it and become habituated or accustomed to the stimulus. Less attention is required from the listener and the voice will fall into the background [4]. The notion of musical surprise is also related to the concepts of 'expectation', 'realisation' and 'denial', as they have been put forth in recent music theories [5, 6]. If we assume that the melody is the musical line that commands most attention and presents most new information, it seems natural to investigate melodic complexity measures as a basis for melody detection algorithms.

Indeed, the idea of using information-theoretic complexity measures to characterise aspects of musical development is not at all new. For instance, to cite just two, in [7], a measure of *Information Rate* [8] computed over a piece of music was shown to correlate in significant ways with familiarity ratings and emotional force response profiles by human human subjects. In [9] it was shown that kernel-based machine learning methods using a compression-based similarity measure on audio features perform very well in automatic musical genre classification.

3 Related Work

Current work of ours [3] indicates that in classical music, the complexity or information content of a sequence of notes may be directly related to the degree to which the note sequence is perceived as being part of the melody. The algorithm described in [3] predicts locally at any point in the music the notes expected to belong to the melody (the algorithm also requires the music to be divided into tracks or voices). The complexity is measured in terms of entropy of notes in the musical surface.

The problem addressed in this paper is somewhat similar. The melody is now expected not to change between the tracks, so a single track must be predicted. This calls for a different prediction strategy. In addition to the local measures of complexity, also global complexity measures based on entropy and compression of entire tracks are examined. A different evaluation data set is required as well – we have tested our hypothesis on popular music, assuming the melody is not likely to change between the tracks.

Melody track identification has recently been examined from a feature extraction and melody/accompaniment learning point of view[10, 11]. Statistical properties of tracks and of note material (pitches, intervals, and note durations) from melody and non-melody tracks can be learned to build a classifier. This approach seems to work quite well. We consider our contribution to be able to fit very well into or extend these models, providing a few, but very significant features.

4 Music Complexity Measures

Shannon's entropy [12] is a measure of randomness or uncertainty in a signal. If the predictability is high, the entropy is low, and vice versa. We will apply this measure to music in a suitable encoding. Let $X = \{x_1, x_2, \dots, x_n\}$ and $p(x) = Pr(X = x)$ then the entropy $H(x)$ is defined as:

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x) \tag{1}$$

X could for example be the set of MIDI pitch numbers and $p(x)$ would then be the probability (estimated by the frequency) of a certain pitch. In the case that only one type of event (one pitch) is present in the current time window, that event is highly predictable or not surprising at all, and the entropy is 0. Entropy is maximised when the probability distribution over the present events is uniform.

4.1 Entropy of Musical Dimensions

We are going to calculate entropy of 'features' extracted from sequences of notes. We will use features related to pitch and duration of the notes. A lot of features are possible: MIDI pitch number, MIDI interval, pitch contour, pitch class, note duration, inter onset interval (IOI) etc. (cf. [13]). We will test the following three basic ones:

1. Pitch class (C): consider the list of notes as a list of *pitch class* events (the term pitch class is used to refer the 'name' of a note, i.e., the pitch irrespective of the octave, such as C, D, etc.);
2. MIDI Interval (I): encode the list of notes as a list of melodic intervals between consequtive notes (e.g., minor second up, major third down, ...);
3. Inter onset interval (O): encode the list of notes in terms of inter onset interval classes, where the classes are derived by discretisation (a IOI is given its own class if it is not within 10% of an existing class).

Each encoding then yields a sequence of events from a given sequence of notes, and entropies can be calculated from the frequencies of these events resulting in the following three basic measures: H_C, H_I, and H_O. Two weighted combinations of the basic features will also be tested: $H_{CO} = \frac{1}{2}(H_C + H_O)$ and $H_{CIO} = \frac{1}{4}(H_C + H_I) + \frac{1}{2}H_O$.

Entropy is also defined for a pair of random variables with joint distribution:

$$H(X,Y) = -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 [p(x,y)] \tag{2}$$

We will test two joint entropy measures: Pitch class in relation to IOI ($H_{C,O}$) and interval together with IOI ($H_{I,O}$). These are expected to be more specific discriminators.

The model is not using information related to performance aspects of the midi file although an actual performance might influence the listener's experience of the piece. What we try to measure is solely the information present in the musical surface.

4.2 Compression by Substitution

The entropy function is a purely statistical measure related to the frequency of events. No relationships between events is measured. For example, the events `abcabcabc` and `abcbcacab` will result in the same entropy value. However, if we were to remember the first string we would probably think of something like three occurrences of the substring `abc` – we infer *structure*. According to Snyder, we perceive music in the most structured way possible [4].

To account for this, complexity measures based on compression could be considered. Music that can be compressed a great deal (in a lossless way) can then be considered less complex than music that cannot be compressed. Shmulevich and Povel [14] have examined methods for measuring the complexity of short rhythmic patterns which are supposed to repeat infinitely. Tanguiane's measure [15] is based on the idea that a rhythmic pattern can be described as elaborations of simpler patterns. Methods exist that substitute recurring patterns with a new event, and store the description of that pattern only once, e.g. run-length encoding or LZW compression [16]. This idea has been discussed in several musical application contexts (e.g., in Music Information Retrieval [9] or in automated music analysis and pattern discovery [17]).

LZW compression is not well suited for compressing short sequences – we will examine compression of entire tracks as a melody prediction method in section 5.3.

5 Prediction Models

We are going to test three prediction models, all capable of producing a melody track prediction, when presented with a MIDI file. The methods will be tested with several encodings of the music, in order to see which variations result in the highest prediction correctness.

First some assumptions about the MIDI files have to be made. A MIDI file can have all events stored on a single track. However, the type of data we are interested in for this experiment are polyphonic pieces that have already been 'streamed' into voices or tracks. Stream separation is a music analysis problem on its own. Although recent work by the authors [18] indicates that this can be solved quite effectively via heuristic search, the MIDI files we use in the current experiments have instruments stored on separate tracks.

The methods we apply to the tracks assume that the tracks are more or less single-voiced (do not contain chords). Methods for reducing a polyphonic track into a representative monophonic sequence of notes where no notes are overlapping in time, are often referred to as *skyline*-algorithms. Some variants are suggested by Uitdenbogerd and Zobel [19]. We do not require total monophony of the tracks, but a simple reduction step is adopted. Notes having onsets separated by no more than 35 ms are assumed to onset at the same time and are treated as a chord. For every chord in every track, only the highest pitched note is taken into account.

5.1 Entropy-based Local Prediction

This method considers the note material through a sliding window. The window is advanced from the beginning to the end in steps of 200 ms. Notes sounding simultaneously with any part of the window are considered to be present in that window.

For each track present in a window, a complexity value calculated on the features extracted from the 'sky-lined' note sequence is calculated (e.g. H_C, entropy of the pitch classes of the events). The track yielding the highest entropy value is the 'winner' of that window. Summing the winners over all windows gives a prediction of which track contains the most complex voice for the longest time. This voice will be predicted as the melody.

Different window sizes of 6, 9, 12, and 15 seconds have been examined, each in combination with the 7 feature encodings presented in section 4.1.

5.2 Entropy-based Global Prediction

A simple variant of the local prediction method is to calculate the entropy of the entire track, and predict the track with the overall greatest complexity. This is also done in the 7 different encodings of the music.

5.3 Compression-based Global Prediction

This model predicts the track that can be compressed the least with an implementation of the LZW algorithm [16].

Sequences of events are transformed into strings of letters – one letter for each event type. The size of the string s before and after compression is recorded, and a compression ratio $r = size(lzw(s))/size(s)$ is calculated. The track resulting in the highest compression ratio – the track believed to be the most complex voice – is predicted as the melody.

Applying the compression algorithm to short strings is likely to actually expand the string ($r > 1.0$), giving misleading results. Simply ignoring all non-compressable tracks proved to be an unfruitful strategy. Instead tracks with less than 100 events were given an artificial ratio of 0.0 – taking them out of competition for the melody selection.

We examine the following encoding possibilities of the events in the tracks: Pitch Class, Interval, IOI, Pitch Class × IOI, and Interval × IOI.

6 Experiments and Results

The prediction algorithms have now been described, and we can run the experiments. We want to test the hypothesis that we tend to listen to the most complex voice at all times, and that this voice is experienced as melody. Popular music in indeed often made in such a way that many accompanying instruments play a pattern or a figure most of the time. All our prediction models are designed to predict the least redundant voice. If the melody really is less repetitive than the accompaniment, our methods will exploit this.

6.1 The Data

The prediction algorithms have been tested on two data sets compiled of MIDI files found on the Internet. The first is a set of popular songs from the 70's to the 90's ('Traditional') – 79 files of pop and rock music hits, film themes etc., (e.g. 'Africa' (Toto), 'Cant Help Falling In Love' (UB40), 'Country Roads' (John Denver), 'Blueberry Hills' (Fats Domino)). The second set ('Modern') contains 29 songs downloaded from an Internet MIDI file download site – all were found among the most popular 50 songs in September 2006 (artists like 50 Cent, Britney Spears, Evanescence, Linkin Park, and Maroon5).

All files were manually annotated by a trained musicologist, and a single track was annotated as the melody (however, in case of more tracks representing the melody in unison, all these tracks were annotated). More files were originally downloaded, but a great deal of files were found to be single track MIDI files, and then discarded. In a few cases cases the melody was found to be shifting so much between different tracks, that the annotator was not able to decide which was the main melody. Such files were also omitted.

The files in the Traditional data set contain each between 3 and 24 tracks – 9.05 tracks on average per file. The Modern data set between 5 and 21 tracks – 11.0 on average. A theoretical baseline for the classification task can be calculated by averaging the number of melodies (in unison) per file divided by the number of tracks per file. This tells us that by random guessing we would be able to achieve 15.3% of the melody tracks correct in the Traditional data set, and 10.6% correct in the other.

6.2 Results

Table 1 shows the prediction results from the classification experiments based on entropy. For each data set, we list local window-based prediction for four window sizes, and the results of global classification via entropy of the entire track. In each experiment (column) the value of the most successful predictor has been highlighted.

The numbers in the last row (Pi) correspond to a baseline experiment using just the average pitch as a measure instead of entropy (predicting the highest pitched voice in each window/track). This simple strategy can compete with

	Correctness (%), Traditional					Correctness (%), Modern				
Measure	6 s	9 s	12 s	15 s	track	6 s	9 s	12 s	15 s	track
H_C	25.3	24.1	29.1	30.4	22.8	27.6	27.6	27.6	27.6	13.8
H_I	27.8	29.1	29.1	26.6	24.1	20.7	20.7	24.1	24.1	10.3
H_O	48.1	**49.4**	**51.9**	**50.6**	34.2	**62.1**	**58.6**	**58.6**	55.2	37.9
H_{CO}	48.1	48.1	48.1	46.8	35.4	55.2	51.7	51.7	51.7	37.9
H_{CIO}	43.0	48.1	49.4	49.4	41.8	51.7	55.2	51.7	51.7	37.9
$H_{C,O}$	32.9	41.8	48.1	49.4	34.2	41.4	48.3	**58.6**	**58.6**	37.9
$H_{I,O}$	30.4	39.2	43.0	43.0	**39.2**	31.0	37.9	51.7	51.7	**41.4**
Pi	**50.6**	45.6	44.3	43.0	36.7	34.5	34.5	34.5	34.5	20.7

Table 1. Entropy-based prediction results

the other strategies in the Traditional data set, but is not of much use when estimating the melody track in the Modern collection.

Table 2 lists the results of the compression based approach. When looking at the tracks globally, LZW compression of events seems to be a better strategy than taking the entropy of the events, which in turn is better than just picking the highest pitched voice – at least under the conditions examined.

Encoding	Correctness (%), Traditional	Correctness (%), Modern
C	32.9	20.7
I	32.9	31.0
O	**43.0**	**51.7**
$H_{C,O}$	39.2	37.9
$H_{I,O}$	39.2	41.4

Table 2. Compression-based prediction results

The most significant finding is that the measures based solely on timing information of the tracks (IOI) are the most successful classifiers. It tells us that there is a strong correlation between rhythmic complexity and melody perception. It could derive from the simple fact that the melody in popular music is strongly related to producing the words of the song, which then might have a more complex emphasis pattern or just rhythmical interpretation than any accompanying instruments.

The melody prediction models are far from perfect. The algorithm is often mislead when there is a solo instrument in the music, that can take over the role of being the instrument we would prefer to listen to for a longer while. Instruments contantly playing small ‘fills’ also attract the attention of our prediction models.

In some songs the accompaniment is simply more important than the melody e.g. when the melody is moving ‘slowly’ by sustaining long notes. Again the lyrics of the song might be an important factor: the melody can be percieved as the most important voice simply because it expresses meaning through words. We are not going to catch this kind of complexity from the MIDI file.

7 Conclusion

Methods for measuring complexity in music were proposed, and used as a basis for melody track prediction models. The different measures and prediction models were tested on two data sets of popular music. The significance of different parameter settings of the models was reported.

Although our models do not comprise the entire truth about the concept of melody, our recognition rates tells us that complexity alone is certainly an important factor. Besides testing our approach on more data, we expect that the way to continue is to combine our research with a statistical approach. Since our approach is not hooked up on any learned average values, we expect it to be a valuable addition to these kind of systems.

8 Acknowledgments

This research was supported by the Viennese Science and Technology Fund (WWTF, project CI010). The Austrian Research Institute for AI acknowledges basic financial support from the Austrian Federal Ministries of Education, Science and Culture and of Transport, Innovation and Technology.

References

1. Weyde, T., Datzko, C.: Efficient melody retrieval with motif contour classes. In: Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005), London, U.K. (2005)
2. Dannenberg, R., Birmingham, W., Pardo, B., Hu, N., Meek, C., Tzanetakis, G.: A comparative evaluation of search techniques for query-by-humming using the musart testbed. Journal of the American Society for Information Science and Technology (2006) in press
3. Madsen, S.T., Widmer, G.: Towards a computational model of melody identification in polyphonic music. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), Hyderabad, India (2007)
4. Snyder, B.: Music and Memory: An Introduction. MIT Press (2000)
5. Narmour, E.: The Analysis and Cognition of Basic Melodic Structures. University of Chicago Press, Chicago, IL (1990)
6. Huron, D.: Sweet Anticipation: Music and the Psychology of Expectation. MIT Press, Cambridge, Massachusetts (2006)
7. Dubnov, S., S.McAdams, Reynolds, R.: Structural and affective aspects of music from statistical audio signal analysis. Journal of the American Society for Information Science and Technology **in press** (2006)
8. Dubnov, S.: Non-gaussian source-filter and independent components generalizations of spectral flatness measure. In: Proceedings of the International Conference on Independent Components Analysis (ICA 2003), Nara, Japan (2003)

9. Li, M., Sleep, R.: Genre classification via an lz78-based string kernel. In: Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005), London, U.K. (2005)
10. Rizo, D., de León, P.J.P., Pertusa, A., Pérez-Sachno, C., nesta, J.M.I.: Melodic track identification in midi files. In: Proceedings of the 19th International FLAIRS Conference, (Melbourne Beach, Florida)
11. Friberg, A., Ahlbäck, S.: A method for recognising the melody in a symbolic polyphonic score (abstract). In: Proceedings of the 9th International Conference on Music Perception and Cognition (ICMPC), (Bologna, Italy)
12. Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal **27** (1948) 379–423, 623–656
13. Conklin, D.: Melodic analysis with segment classes. Machine Learning **Special Issue on Machine Learning in Music**(in press) (2006)
14. Shmulevich, I., Povel, D.J.: Measures of temporal pattern complexity. Journal of New Music Research **29**(1) (2000) 61–69
15. Tanguiane, A.: Artificial Perception and Music Recognition. Springer, Berlin (1993)
16. Ziv, J., Lempel, A.: A universal algorithm for sequential data compression. IEEE Transactions on In-formation Theory **23**(3) (1977) 337–343
17. Lartillot, O.: A musical pattern discovery system founded on a modeling of listening strategies. Computer Music Journal **28**(3) (2006) 53–67
18. Madsen, S.T., Widmer, G.: Separating voices in midi. In: Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006), Victoria, Canada (2006)
19. Uitdenbogerd, A.L., Zobel, J.: Manipulation of music for melody matching. In: ACM Multimedia. (1998) 235–240