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 of Computational Perception, Johannes Kepler University, Linz
gerhard.widmer@jku.at

Abstract. In this paper, we will examine 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 – that it will be the least redundant voice. Measures of melodic complexity calculated from pitch and timing information 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 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 (unconsciously) picking out those notes in a – possibly complex – multi-voiced piece that constitute the melodic line. Current work of ours describes an automatic method for locating the notes constituting a likely melody throughout a piece of classical music stored in a MIDI file [1]. Reflecting the fact that the melody can change between the voices present, the algorithm is able to construct the melody of predicted notes from different voices in the music.

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. The goal is to evaluate the importance of one single factor in solving this problem: music complexity. Different complexity measures are proposed and evaluated.

Melody track identification is useful in systems that indent 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. Melody is also an important aspect in music-related computer applications, for instance, in Music Information

* In Proceedings of the International Workshop on Artificial Intelligence and Music (MUSIC-AI 2007) held in conjunction with the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), Hyderabad, India, 6-12 January 2007.

Retrieval (e.g., in music databases that offer retrieval by melodic motifs [2] or Query by Humming [3]).

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 concept of 'expectation' as it has 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 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 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 [1]. The algorithm described predicts at any point in the music the notes expected to belong to the melody by comparing the complexity of each voice, when looking locally at the immediately preceding notes (the algorithm 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 melody track identification 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 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 less likely to change between the tracks.

Melody track identification has recently been examined as a melody/accompaniment classification problem [10, 11]. Statistical properties (features) of tracks

and of note material (pitches, intervals, and note durations) from melody and non-melody tracks can be learned and used 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, by providing a few 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 be a discrete random variable on a finite set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ with probability distribution $p(x) = Pr(X = x)$. Then the entropy $H(X)$ of X is defined as:

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

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 (PC): 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 (Int): encode the list of notes as a list of melodic intervals between consecutive notes (e.g., minor second up, major third down, ...);
3. Inter onset interval (IOI): 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_{PC}, H_{Int}, and H_{IOI}. Two weighted combinations of the basic features will also be tested: $H_{PC,IOI} = \frac{1}{2}H_{PC} + \frac{1}{2}H_{IOI}$ and $H_{PC,Int,IOI} = \frac{1}{4}(H_{PC} + H_{Int}) + \frac{1}{2}H_{IOI}$.

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

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

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

4.2 Compression by Substitution

The entropy function is a purely statistical measure related to the frequency of events. No relationships between events are 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, which will predict a melody track, 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.

Some assumptions about MIDI files have to be made. We require voice information to be present in the file. MIDI files can be structured into tracks where each track contains the events from an instrument (voice).[3]

If all events belong to the same voice, the melody *track* prediction task is trivial. However, the more general problem of extracting the melody from such

[3] Voice information can also be encoded as events belonging to different channels within a track.

files is a hard, but interesting one, which might require automatic stream separation (adding voice information to the events) before predicting the melody. Stream separation is a music analysis problem on its own. Although recent work of ours indicates that this can be solved quite effectively via heuristic search [18], we will use files containing voice information, in order not to mix different factors in our investigation. After all, the voice information is part of the ground truth of the experiment.

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]. Stream separation might also be useful for this task. 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_{PC}, 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 an estimate 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 will be 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 all events in each track, and predict the track with the overall greatest complexity. This will also be 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-compressible tracks proved to be an unfruitful strategy. Instead tracks with less than 100 events were given an artificial ratio of zero – 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 $\times$ IOI, and Interval $\times$ IOI.

6 Experiments and Results

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), 'Can't Help Falling In Love' (UB40), 'Country Roads' (John Denver), 'Blueberry Hill' (Fats Domino)). The second set ('Modern') contains 29 songs downloaded from an Internet MIDI file download site – all were found among the most popular 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 some were found not to contain voice information and then discarded as trivial (as discussed in section 5). In a few 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. It was decided to omit such files since they contain ambiguous ground truth.

The files in the Traditional data set contain each between 3 and 24 tracks with notes – 9.05 on average per file. The files in the Modern data set contain each between 5 and 21 tracks with notes – 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

The prediction algorithms are evaluated in terms of percentages of the files in the data set that had the melody track correctly predicted.

Table 1 shows the prediction results from the classification experiments based on entropy. For each data set, we list local window-based prediction (see 5.1) for four window sizes, and the results of the global classification via entropy of the entire track (see 5.2). In each experiment (column) the value of the most successful predictor has been highlighted.

	Correctness (%), Traditional					Correctness (%), Modern				
Measure	6 s	9 s	12 s	15 s	track	6 s	9 s	12 s	15 s	track
H_{PC}	25.3	24.1	29.1	30.4	22.8	27.6	27.6	27.6	27.6	13.8
H_{Int}	27.8	29.1	29.1	26.6	24.1	20.7	20.7	24.1	24.1	10.3
H_{IOI}	48.1	**49.4**	**51.9**	**50.6**	34.2	**62.1**	**58.6**	**58.6**	55.2	37.9
$H_{PC,IOI}$	48.1	48.1	48.1	46.8	35.4	55.2	51.7	51.7	51.7	37.9
$H_{PC,Int,IOI}$	43.0	48.1	49.4	49.4	**41.8**	51.7	55.2	51.7	51.7	37.9
$H_{PC\times IOI}$	32.9	41.8	48.1	49.4	34.2	41.4	48.3	**58.6**	**58.6**	37.9
$H_{Int\times IOI}$	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. Correctness is the percentage of the files in the data set that had the melody track correctly predicted by using the respective measure.

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 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 explained in 5.3. 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
PC	32.9	20.7
Int	32.9	31.0
IOI	**43.0**	**51.7**
$PC \times IOI$	39.2	37.9
$Int \times IOI$	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 suggests 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 algorithm is often misled when there is a solo in the music, that can take over the role of being an alternative melody for a longer while. Instruments constantly playing small 'fills' also attract the attention of our prediction models. In such cases, it is not our assumption that is wrong, but the evaluation method that is too coarse.

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 perceived 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 (actually it assumes almost no musical background knowledge), 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. By using discriminative machine learning, we can train a classifier to optimise the contribution of complexity and other features (like average pitch) in a melody track prediction task. Since our approach is not dependent up on any learned 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. 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)

2. 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)
3. 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
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 **57**(11) (2006) 1526–1536
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., Iñesta, J.M.: Melodic track identification in MIDI files. In: Proceedings of the 19th International FLAIRS Conference, Melbourne Beach, Florida (2006)
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 (2006)
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 Information 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