MIDI Music Genre Classification by Invariant Features

Adi Ruppin **Hezy Yeshurun**
School of Computer Science
Tel Aviv University, Israel
ruppin@att.biz, hezy@math.tau.ac.il

Abstract

MIDI music genre classification methods are largely based on generic text classification techniques. We attempt to leverage music domain knowledge in order to improve classification results.

We combine techniques of selection and extraction of musically invariant features with classification using *compression distance* similarity metric, which is an approximation of the theoretical, yet computationally intractable, Kolmogorov complexity.

We introduce several methods for extracting features which are invariant under certain transformations commonly found in music. These methods, combined with data compression, generate a lossy compressed representation which attempts to preserve feature invariance. We analyze the performance of each method, thus gaining insight into the features that are significant to the human perception of music.

Keywords: Genre classification, Kolmogorov complexity

1. Introduction

We seek to extract features which are the basic musical building blocks, and widely reoccur within a musical piece or genre, often undergoing certain transformations. Composers, psychologists and researchers place great importance on such features. Lehrdal and Schenker [7, 13] have identified significant repetition as essential to the interpretation of music. We aim to process our corpus in such a way as to preserve features invariance under such transformations.

For classification we use compression distance [4] to measure similarity between the musical pieces. In addition to this method's power, compression effectively captures repeating patterns.

2. Related work

Common classification methods include Baysean classifiers, Decisions Trees, Neural Networks and Hidden Markov Chains [8]. These have primarily been used in text classification.

Works pertaining specifically to music mostly deal with audio signals [14]. MIDI classification works include statistical methods, neural networks techniques [5], pitch class methods [2], multi-resolution views [6], self organizing networks [1], clustering according to compression distance [4] as well as other approaches.

3. Method

We represent music as a time function expressing n-dimensional pitch and duration vectors (chords):

$$f(t): Z \rightarrow (Z^n \times \Re^n) \qquad (1)$$

3.1 The transformations

We define the transformations commonly used in music:

3.1.1 Transposition transformation

Transposition occurs frequently in music and involves a theme or segment being played at a constant pitch offset:

$$g(t) = f(t) + [c \cdot \vec{1}, \vec{0}]^T, t \in [a,b] \qquad (2)$$

We would like extracted pitch features to be invariant to this transformation, as two musical pieces or segments can be considered equivalent when played at a different pitch.

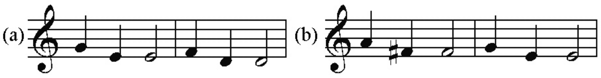


Figure 1: Folk song transposition sample

3.1.2 Augmentation/diminution transformation

A musical theme or segment is often played or repeated at a different speed, or tempo:

$$g(t) = f(t) \cdot [\vec{1}, \lambda\vec{1}]^T, t \in [a,b] \qquad (3)$$

Two such segments which differ only in tempo or by a fixed note length ratio are considered an *augmentation* or *diminution* and can be considered equivalent.

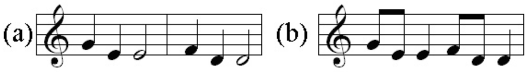


Figure 2: Folk song diminution sample

3.1.3 Sequential modulation transformation

Sequential modulation, or “inexact” modulation, does not preserve exact pitch distances, typically introducing no more than a small error (1/2 tone):

$$g(t) = f(t) + [\vec{C} + \vec{C}_t, \vec{0}]^T, C_{t,i} \in \{0, \frac{1}{2}, -\frac{1}{2}\} \qquad (4)$$

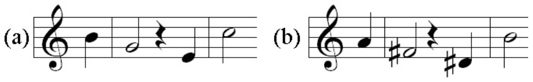


Figure 3: Sequential modulation sample (excerpt from Brahms Symphony No. IV)

3.1.4 Crab transformation (“crab form”):

Crab form inverts the pitch for a melodic segment:

$$g(t) = [\vec{C}, \vec{0}] - f(t) \cdot [\vec{1}, -\vec{1}]^T, t \in [a,b] \qquad (5)$$

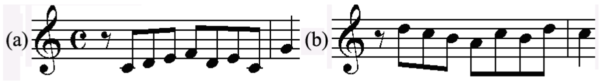


Figure 4: Crab sample (excerpts from Bach Prelude #1)

3.2 The method, step-by-step

3.2.1 Step 1

First, we take the quantized melody contour, ignoring MIDI note-off and other events and accepting only note-on events, thus disposing of performance-sensitive data.

To ensure invariance to transformations 3.1.1 and 3.1.2, we take the derivatives of the pitch and duration $\frac{d}{dt} f(t)$ (we look at the pitch and time differences and dispose of the absolute pitch and time). Note duration change is more effectively treated by calculating the time *ratios*, so we derive the logarithm of the duration.

For inexact sequentials described in 3.1.3 an optional preprocessing step truncates the exact pitch intervals and denotes only pitch direction (up or down) and whether it is a step (1/2 or 1 tone) or a jump (1.5 tones or higher).

Crab form described in 3.1.4 can be treated by preserving only the change of the pitch direction instead of the absolute pitch direction (the second derivative).

For the purpose of comparison with earlier works, we also experiment with extracting the pitch normalized to its difference from the musical piece’s average pitch.

Finally, we observe that musical pieces may include patterns repeating at different hierarchies. For example, a musically significant repetition may occur at f(kt) a<t<b. To uncover such underlying musical structure, multi-resolution methods are applied. In our case, we applied a simple low-pass filter.

3.2.2 Step 2

The compression distance is a metric [4] which does not rely only on a small set of features, but rather attempts to calculate an ideal *information distance*. According to this principle, two objects are considered close if each can be well compressed using the information provided by the other object:

$$d(x,y) = \frac{\max\{K(x \mid y), K(y \mid x)\}}{\max\{(K(x), K(y)\}} \qquad (7)$$

Where K represents Kolmogorov complexity.

Compression may be viewed as modeling the more ideal *Kolmogorov complexity*[1]. The caveat to the Kolmogorov distance approach is that it is not computable. For this reason we can only attempt to approximate it by a standard compression algorithm such as Lempel Ziv [15]. We remove from the compression short patterns that are likely to reoccur and skew the results, typically sequences shorter than 4 or 5 symbols.

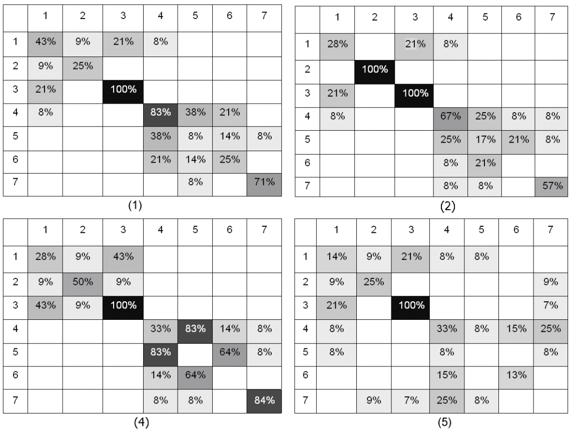
3.2.3 Step 3

We perform classification using the k-NN algorithm, which takes into account k musical pieces to determine which category is closest. We typically take k=3.

4. Experiments

The test collection was comprised of 50 musical pieces in MIDI format, from 3 main categories: classical music, pop music and traditional Japanese music. The first two genres are further divided by composer: Mozart, Brahms, Vivaldi and the Beatles, Abba and Britney Spears respectively. We perform the following experiments using Leave-one-out cross validation: (1) We take the pitch/time derivative; (2) We take the pitch/time derivative and truncate exact intervals; (3) The pitch/time derivative after a low-pass filter; (4) The pitch/time second derivative; and (5) The average pitch method.

Some confusion matrices for the above experiments are shown below. For each category, scores indicate the number of matches out of the total elements in classes.



	1	2	3	4	5	6	7
1	43%	9%	21%	8%			
2	9%	25%					
3	21%		100%				
4	8%			83%	38%	21%	
5				38%	8%	14%	8%
6				21%	14%	25%	
7					8%		71%

(1)

	1	2	3	4	5	6	7
1	28%		21%	8%			
2		100%					
3	21%		100%				
4	8%			67%	25%	8%	8%
5				25%	17%	21%	8%
6				8%	21%		
7				8%	8%		57%

(2)

	1	2	3	4	5	6	7
1	28%	9%	43%				
2	9%	50%	9%				
3	43%	9%	100%				
4				33%	83%	14%	8%
5				83%		64%	8%
6				14%	64%		
7				8%	8%		84%

(4)

	1	2	3	4	5	6	7
1	14%	9%	21%	8%	8%		
2	9%	25%					9%
3	21%		100%				7%
4	8%			33%	8%	15%	25%
5	8%			8%			8%
6				15%		13%	
7		9%	7%	25%	8%		

(5)

Figure 5: Sample confusion matrices from experiments

[1] Kolmogorov complexity K(x) is defined as the length of the shortest compressed binary version from which x can be fully reproduced

Overall, in selecting invariant features we see a tradeoff between the quality of an exact composer match and the quality of a more general genre match. The degree of “lossiness” influences classification accuracy for different corpora. For example, Vivaldi is recognized almost perfectly by all methods as it is very structured, however Brahms is not classified correctly by method (1) but is recognized by (2). On closer inspection, these pieces include many sequential modulations (inexact repetitions), that are largely missed by methods which rely on exact intervals. On the other hand, method (2) failed in other genres, as it is more error prone.

Table 1: Results summary

	1	2	3	4	5
Composer match	51%	51%	58%	42%	27%
Match or 2nd label	73%	57%	62%	62%	45%
Genre match	80%	75%	72%	85%	58%

5. Comparison with other methods

Below is a performance summary of different methods.

MIDI/ Audio	By	Date	Method	Genres	Corpus size	Success
M	Chai & Vercoe	2001	HMM	3	491	63%
M	Ponce de Leon	2002	SOM	2	N/A	77%
M	Shan & Kuo	2003	Associative classification	2	70-100	84%
M	McKay	2004	NN	3	255	84%
M	Cilibrasi	2004	Compression	3	36	80%
M	Lin	2004	Repetitions	7	500	49%
M	Pollastri	2001	HMM	5	100	49%
A	Li & Tzanetakis	2003	LDA	10	1000	71%
A	Tzanetakis & Essl	2001	Gaussian	6	300	62%
A	Burred	2003	GMM	13	850	60%

Figure 6: Comparison with other methods

6. Conclusion and future work

Even with simple compression such as LZW good results were obtained, possibly rivaling those achieved by humans [11]. Since LZW essentially eliminates continuous repetitions, we can conclude that repetition in music occurs more often than the human ear might recognize and is instrumental for its classification. This is consistent with music theory notion of an underlying repetitive structure.

Our invariant features approach produces better classification results than most existing methods, such as the pitch-averaging method. The performance of the different methods highlights what features are applicable to various corpora. Still, for best results some manual fine-tuning is required, as one method may be more fitting a specific corpus than another.

Future work may include experiments with additional compression algorithms, possibly capable of handling the multi-resolution nature of music, such as DCT. Additionally, additional clustering techniques should be explored.

7. References

[1] C. Anagnostopoulou, G., Westermann, “Classification in Music: A Computational Model for Paradigmatic Analysis”, in *Proc. of the International Computer Music Conference*, 1997.

[2] S, Blackburn, and D. De Roure, “Musical Part Classification in Content Based Systems”, in *Proc. of ACM Multimedia ’98*, 1998, pp. 361-368.

[3] W. Chai, B. Vercoe, Folk music classification using hidden Markov models. *In Proc. of International Conference on Artificial Intelligence*, 2001.

[4] R. Cilibrasi, P. Vitanyi, R. De Wolf, “Algorithmic Clustering of Music”, [Web site] Available: http://xxx.lanl.gov/abs/cs.SD/0303025.

[5] R. Dannenberg, B. Thom, D. Watson, ``A Machine Learning Approach to Musical Style Recognition" in *1997 International Computer Music Conference*, International Computer Music Association, pp. 344-347.

[6] E.W. Large, C. Palmer, J.B. Pollack, “Reduced Memory Representations for Music”, *Cognitive Science*, 1995, pp. 53-96.

[7] F. Lerhdal, R. Jackendoff, “A Generative Theory of Tonal Music”, Cambridge MIT press, 1983.

[8] Y.H. Li., A.K. Kain, Classification of Text Documents, *Computation Journal*. 41, 8, pp. 537-546.

[9] C.R. Lin, N.H. Liu, Y.H. Wu, L.P. Chen, “Music Classification Using Significant Repeating Patterns”, *DASFAA*, 2004: pp. 506-518

[10] C. McKay, I. Fujinaga, “Automatic Genre Classification Using Large High-Level Musical Feature Sets”, in *5th International Conference on Music Information Retrieval*, 2004.

[11] E. Pollastri, G. Simoncelli, “Classification of Melodies by Composer with Hidden Markov Models”, in *Proc. of the international conference on Web delivering of music*, 2001.

[12] P.J. Ponce-de-León, Musical style identification using self-organizing maps, in *Proc. of International Conference on Web Delivering of Music*, Wedelmusic 2002. IEEE Computer Society Press, pp. 82-92, 2002.

[13] H. Schenker, “Harmony”, Edited by Oswald Jonas, translated by Elisabeth Mann Borgese. Cambridge: M.I.T. Press, 1954.

[14] G. Tzanetakis, P. Cook, “Automatic Music Genre Classification of Audio Signals”, in *Proc. Of Int’l Symposium on Music Information Retrieval*, 2001.

[15] J. Ziv, A. Lempel, “A Universal Algorithm for Sequential Data Compression”, *IEEE transactions on information theory*, 1997.