Understanding Music from Symbolic Representation with Higher Order Networks

Xindi Wang* Ricky Laishram†

Abstract
Simple networks have been used to model and analyze music. However, this assumes Markov property – there is no higher order memory and dependencies. In this work, we propose using higher order networks `MusicHON` to model music from the symbolic representation, which incorporates higher-order dependencies between notes. We propose various music-related features that can be extracted from this representation. Feature distribution and Principal Component Analysis show that these features provide insights on music genres and artists. We further demonstrate the power of `MusicHON` and its features through music genre classification. We find that `MusicHON` outperforms the simple network baseline significantly. Feature importance analysis of the classifier additionally shows that the `MusicHON` features are meaningful and align with common perception of music genres. We show that `MusicHON` – a higher order network representation of music – is more accurate, and provides more insights about music than simple networks.

Keywords— higher order networks, music, classification

1 Introduction

Music is a natural complex system: notes interact sequentially to create melody and interact vertically to create harmony. With only 12 notes (in western music), composers are able to create different styles and music genres.

Because of the huge variety and quantity of music that is available nowadays, analysis and classification of music using data mining techniques is an area of great research interest. It helps us in understanding how music develops over time, characteristics of different music genres, etc. at a larger scale.

Analysis of music through data mining generally falls into two categories based on how the music data is represented – symbolic and audio. In the former case, each music piece is represented by the music notation (e.g. sheet music); and for the latter, each piece is represented by an audio wave. In this paper, we focus on the symbolic representation.

Various methods have been proposed to analyze music from the symbolic representation. There are various researches classifying music genres from the symbolic representation of the pieces [15, 1, 16] where features such as key number, duration, pitch are extracted directly from the symbolic representation and used for classification. Simple networks and fixed low order networks have also been used to model and analyze music [23, 9, 5]. These representations assume there are no or limited dependencies between notes.

In contrast, in this paper, we take the higher-order dependencies between notes into account and model music piece as a *Higher Order Network (HON)* [28]. We propose several features that can be extracted from HON. From the feature distributions, we observe that these features align with the common perception of music genre. Using Principal Component Analysis on the features, we can spot differences and relationships between different genres and artists. Classification of genre using these features outperforms the ones using simple networks, showing its potential to capture music characteristics. We further perform feature importance analysis to show that these features are insightful.

The contributions of this paper are:

1. We model music pieces as higher order networks `MusicHON` and propose several music-related features that can be extracted from this representation.
2. Through feature distribution and Principal Component Analysis, we are able to gain instights of differences and relationships between music genres and artists.
3. We show experimentally that classifying music genres using features on `MusicHON` outperforms simple network baselines. Further feature importance analysis from the classifier shows how these features align with common music perception.

2 Related Works

2.1 Music Representation Most works on music analysis using data mining techniques can be grouped into two categories - using symbolic representation or audio representation, depending on how the music is represented [7, 27]. In *symbolic representation* the music pieces are represented by sequences of symbols

*Northeastern University, Boston, MA, USA (Email: wang.xind@husky.neu.edu)

†Syracuse University, Syracuse, NY, USA (Email: rlaishra@syr.edu)

that represent note, pitch, time, timbre, etc. In *audio representation*, the music pieces are represented as audio waves – resulting from the recording of sound sources or from direct electronic synthesis.

There are fundamental differences between symbolic representation and audio representation [27]. Symbolic representation is content-aware and describes events in relation to formalized concepts of music (music theory), whereas the audio representation is content-unaware. On the other hand, audio representation is a continuous flow of information, both in time and amplitude, whereas the symbolic representation accounts for discrete events. Since our work is to understand music, we choose the content-aware symbolic representation.

2.2 Symbolic Representation With MIDI MIDI is one of the most common formats of symbolic representation. MIDI (short for Musical Instrument Digital Interface) is a technical standard that describes a communications protocol, digital interface, and electrical connectors that connect a wide variety of electronic musical instruments, computers, and related audio devices for playing, editing and recording music. [24]. MIDI carries event messages, including sheet music information such as a note's notation, pitch, duration as well as performance information such as velocity, vibrato, panning to the right or left of stereo, and clock signals (which set tempo). On the other hand, this format takes a lot less space, which makes it much easier to store and communicate, and allows for better comparison between music pieces played on different instruments. The richness of MIDI data as well as it being computational-friendly makes it a suitable format for large scale music analysis. Various studies have been conducted using MIDI data, including analyzing music similarity [10, 2], music genre classification [12, 4] and even creating computer-composed music [17].

2.3 Representing Music as Simple Networks Simple networks have been used to model and analyze music, where notes are represented as nodes and transition between notes (i.e. change from one note to another) is denoted by edges. Ferreti [9] used the simple network model of music and investigated various network properties including degree, betweenness centrality, modularity, etc. It was found that these music networks are scale free and exhibits the small world property. Serrà et al. [23] analyzed music pieces across time by creating a network of pitch transitions. They used this representation to study the change note frequency distribution over time for western contemporary popular music. Corrêa et al. [5] further includes second order dependencies in music representation separately from the first order dependencies, through building a similarity network, they are able to find community structures of music genres.

2.4 Higher Order Networks Trajectories, as a common real-world data type, have been studied using networks. A common practice for network construction from trajectory data is to build a simple network: two sequentially adjacent entities are connected with an edge, and the number of times of the two entities appearing sequentially as the edge weight [8, 18, 14]. This assumes Markov property (first-order dependency): there is no higher order memory and dependencies in the sequence, which is not always the case in real-world data. To tackle this problem, *Higher Order Networks (HON)* have been proposed to capture higher-order dependencies in data, especially for sequential data. There are various models for higher order networks. The simplest is fixed-order network [19, 11, 22, 21] where different order of Markov transitions are considered. This is easy to model but has the drawback of not considering different order as a whole as well as having potential redundant information. As an improvement, Scholtes et al. develops an algorithm to determine the maximum order needed and represent it as multiple higher-order models up to a maximum order [22]. To tackle redundancy and incorporate multiple orders more flexibly, Xu et al. proposed a model where one generates significant higher-order rules from the data and build a single network using those extracted multi-order rules [28]. Additionally, traditional network analysis tools can be applied on this network directly.

Studies have found that higher-order networks are able to reveal properties that the simple network fails to capture. For example, Rosvall et al. revealed that higher-order information can effect the results of community detection, ranking and spreading process on the network [19]. Xu et al. found that using their higher-order network model, the clustering on global shipping trajectory network are no longer limited to geography distance, with the potential application on identifying regions wherein aquatic species invasions are likely to happen [28]. These results show that higher-order networks are powerful to model the non-Markovian relationship between entities which simple network models lack.

3 Music as Higher Order Networks

In this section we describe how we process a MIDI file to a note sequence (Section 3.1) and use that to create a HON representation (Section 3.2). Then we use these HON to extract features (Section 3.3) that are related to music perception.

3.1 Data and Data Processing

Data Our MIDI corpus is from "Largest MIDI Collection on the Internet".[1] This dataset contains genre labels for each music piece. The genres included are Classical (1751 pieces), Folk (5115 pieces), Jazz (245 pieces), Pop (193 pieces) and Rock (877 pieces) music.

Data Processing The data processing converts the MIDI file for each music piece into a note sequence further fed in the HON generation algorithm. There are four steps involved in data processing: (1) converting MIDI files into note sequences with raw coding, (2) converting the raw coding into relative coding, (3) handling the rests and (4) selecting track.

The MIDI files are processed using `music21`[6] into note sequences with separate tracks. Under the raw MIDI coding scheme, the notes are numbered 0 to 127, representing 12 notes across 10 octaves.

The raw coding scheme is not ideal because the same music piece can be played under different keys. We use *relative pitch coding* to tackle this problem. To obtain the relative pitch coding, we detect the tone note of the piece using `music21`. The relative coding is then obtained through subtracting the tone note code from the raw code.

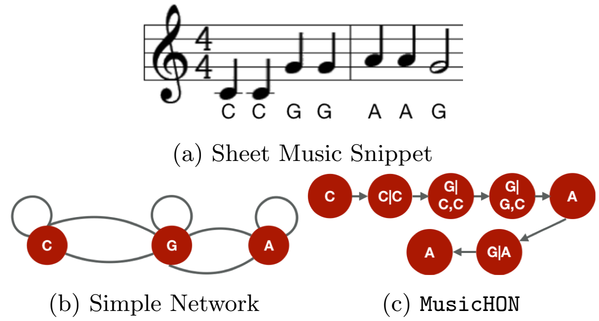
Besides notes, rests are also important in music. However, we observe that the extracted sequence have some rests that are not in the original sheet music. To correct this, we obtain the duration of all the notes in a piece. The shortest note is set as the lower threshold, and the longest as the upper threshold. Rests whose duration are less than the lower threshold are ignored. Of the remaining rests, those with duration less than the upper threshold are considered "short rests" and coded with 128. The remaining rests are considered as "long rests" and denoted by 129 in the sequence.

In MIDI representation, different instruments are denoted by different tracks. In our work, we select the track with the most number of notes for each music piece, with the assumption that this is most likely to be the melody line. After applying the above process, we obtain the note sequence for each music piece coded from 0 to 129.

3.2 Representing Music as Higher Order Networks

In this paper, we represent different music pieces as *Higher Order Networks* (HON) using the method by Xu et al. [28]. In HON, nodes are not always a single entity (note), they can be a sequence of entities that represent dependencies between the entities. The construction of the HON consists of two steps: 1) rule extraction identifies higher-order dependencies of high confidence and sufficient support and 2) network wiring which converts the rules into a graph representation. Using this algorithm, we convert the note sequence of a music piece (obtained through the process in Section 3.1) to HON and we refer this representation as `MusicHON` for further discussion.

[1]`https://www.reddit.com/r/WeAreTheMusicMakers/comments/3ajwe4/the_largest_midi_collection_on_the_internet/`



(a) Sheet Music Snippet

(b) Simple Network

(c) `MusicHON`

Figure 1: **Sheet music snippet of "Twinkle, Twinkle, Little Star" and its simple network and `MusicHON` construction.**

`MusicHON` $H(V, E)$ is a directed weighted network. Node $i \in V$ represents a sequence of notes n, written as $n_c|n_p^1, n_p^2, ...n_p^l$, which means that the current note is n_c and the previous notes are $n_p^1, n_p^2, ...n_p^k$, where k is the number of previous notes. Edge $(i, j) \in E$ is a weighted edge where the weight is the transition probability from node i to j.

As an example, consider the snippet of "Twinkle, Twinkle Little Star" shown in Fig. 1a. The rule from the first two notes is $C \rightarrow C$. Following this rules, the additional rules when we consider the next note in the sequence are $C \rightarrow G$ and $C|C \rightarrow G$. All the possible rules are similarly generated from the note sequence. The rules are filtered based on confidence and support value calculating using the entire music piece. Remaining rules become the edges for `MusicHON`, and the transition probability for these edges are calculated from the note sequence. Fig. 1b and 1c shows the simple network and `MusicHON` representation for "Twinkle, Twinkle Little Star" snippet separately. We can observe that `MusicHON` is able to preserve dependency information of the music itself in contrast to the simple network representation.

3.3 Feature Extraction

The `MusicHON` representation allows us to extract features that are meaningful from a musicology point of view, which might not be applicable under simple networks. We propose the following features:

Abruptness *Abruptness* captures the abrupt changes in music, for example, transition from the verse to the chorus in pop music. We consider these abrupt changes as the infrequent bridges between different sections of the music piece and define abruptness as the *pitch range across the edge that has the highest betweenness centrality relative to the transition probability.*

Denoting $BC(i,j)$ and $p(i,j)$ as the edge betweenness centrality and transition probability of an edge (i,j), and $(i,j)^*$ as the edge such that,

$$(i,j)^* = \arg\max_{(i,j)\in E} \frac{BC(i,j)}{p(i,j)}$$

Denoting the set of pitches of the notes in the higher-order node i by F_i, then the pitch range across an edge (i,j) can be calculated as $\delta(i,j) = \max(\max(F_i) - \min(F_j), \max(F_j) - \min(F_i))$. Finally, we formally define the abruptness of the `MusicHON` by,

$$\mathcal{F}_A = \delta(i,j)^*.$$

Branching The `MusicHON` of some pieces can look more complicated than others. For example, for some classical music, there can be a lot of potential follow-up notes from one note as the starting point. This will make the `MusicHON` have a more "complicated" structure – nodes on average have higher out-degree. In contrast, pop music usually has a distinguishable melody that repeats throughout the piece, resulting in more "chains" in its `MusicHON` and nodes on average have lower out-degree. Therefore, to capture such differences, we define branching as the *average out-degree after removing edges with low transition probability (with a threshold)*. Let $N_o(i)$ be the node i's out-neighbors in H, and θ be the transition probability threshold. Then the branching of `MusicHON` is given by

$$\mathcal{F}_B(\theta) = \frac{1}{|V|}\sum_{i\in V} |\{j \in N_o(i) : p(i,j) \geq \theta\}|.$$

Melodic Melody is a note sequence that repeats throughout the music. Under `MusicHON`, melody should be captured in the higher order nodes. Therefore, we define melodic as the *average length of the higher order nodes in the `MusicHON`*, written as:

$$\mathcal{F}_M = \frac{1}{|V|}\sum_{i\in V} k(i) + 1$$

Repeatedness **(XW: needs more work)** Pop and rock music often have more repeated structures than classical or jazz music. Here we define repeatedness at path length l as *the variance of the random walker probabilities encountering all simple paths with length l*. The intuition is that for a given path length l, if there are paths that are more likely to be traversed, this indicates there are repetitive patterns; if all paths have equal traversing probability, there is no significant repetitive pattern. Denoting p as a simple path in `MusicHON`, and $P(p)$ is the probability of a random walker traversing path p, we can calculate repeatedness as:

$$\mathcal{F}_{R|l} = \operatorname*{Var}_{|p|=l}\{P(p)\}.$$

Pitch Range Pitch range changes with development of instruments over time, composition tradition change – especially on dynamics of music, etc., therefore this range may capture music characteristics over time. In this project, we use three versions of pitch range:

Pitch range in the piece $\mathcal{F}_{P0}$ This is simply the highest pitch minus the lowest pitch in the piece.

Average pitch range in node $\mathcal{F}_{P1}$ Denoting the set of pitches of the notes in the higher-order node i by F_i, we can calculate the pitch range of a node as $\delta_i = \max(F_i) - \min(F_i)$. Then we can calculate the average pitch range in node as

$$\mathcal{F}_{P1} = \frac{1}{|V|}\sum_{i\in V}\delta_i.$$

Average pitch range across edge $\mathcal{F}_{P2}$ Denoting the set of pitches of the notes in the higher-order node i by F_i, the pitch range across an edge (i,j) can be calculated as $\delta(i,j) = \max(\max(F_i) - \min(F_j), \max(F_j) - \min(F_i))$. Then we can calculate the average pitch range across edge as

$$\mathcal{F}_{P2} = \frac{1}{|E|}\sum_{(i,j)\in E}\delta_{i,j}.$$

4 Analysis and Experiments

In this section, we perform analysis and experiments to evaluate whether the features proposed in Section 3.3 can successfully capture the differences in human perception of music genres, and their prediction power on music genre classification. We also perform a case study to look into the relationship between a music piece and its `MusicHON`.

4.1 Feature Distribution To investigate the differences between music genres, we plot the distribution of the features for different genres, shown in Fig. 2.

We observe that for abruptness, Pop, Rock and Jazz music are more "abrupt" than Folk and Classical music. This may be due to that Pop and Rock in general have verse-and-chorus structure, creating a certain level of "abruptness" in the music; Jazz contains a lot of improvisations, which might lead to more abrupt changes.

For branching, we observe that Classical has significantly more "branches" than other genres, following by Jazz and Folk music. Pop and Rock has lower branching scores. This aligns with our expectation that in classical music pieces, there are more possibility from a starting note; where in Pop and Rock, there are clearer "chain" structure in music because of distinctive melody lines.

For repeateness, we show the distribution of $\mathcal{F}_{R|l=5}$ for different genres as well as the median score of $\mathcal{F}_{R|l}$, $l \in [1, 2, 3, 4, 5]$ for each genre. For the distribution of $\mathcal{F}_{R|l=5}$ we can see that Classical has lower $\mathcal{F}_{R|l=5}$ than other genres. Folk, Pop and Rock has higher repeatedness score, which aligns with our expectation. For different l, we observe similar trends of Classical having the lowest repeatedness score, followed by Jazz; Folk, Pop and Rock have higher repeatedness score.

For melodic, we observe that jazz is significantly less "melodic" than other genres. This is very much expected because of the improvisations in Jazz will make it sounds less "melodic".

For pitch ranges, we observe limited difference between music genres in pitch range in piece. However, for pitch range in node and pitch range in edges, we observe that Classical and Folk are significantly lower than other genres. This might be due to that the traditional composition of Classical and Folk tend to have less dramatic changes locally (i.e. there are less dramatic pitch changes within a short distance). In general, the observations of the feature distributions aligns with our common perception of music genres.

4.2 Principal Component Analysis Additional to distribution comparison, we conduct Principal Component Analysis (PCA) [13] on the features we proposed. Since the features have different scale, we standardized each feature before performing PCA. We analyzed PCA on different music genres, as well as selected artists: Bach, Beatles, Mozart, Nirvana and Vivaldi. We extract principle components that corresponds to at least 80% of the variance. We found that three components are enough to satisfy this criteria in all cases.

We plot in Fig. 3 the first two principal components.[2] Each point in the plot is a piece, with the color denoting the music genre or the artist. We draw an ellipse covering 95% of the points for each genre or artist to visually show the clusters.

From Fig. 3(a) we can see that, though all genres overlaps, clusters have different characteristics. For example, the centroid for Classical cluster is further away from other genres; Folk has a tighter cluster while the cluster for Jazz is loose. For different artist (Fig. 3 (b)) we can see that Bach overlaps more with Vivaldi than to Mozart. This may reflect that Bach and Vivaldi are both Baroque era composers and Bach was deeply influenced by Vivaldi [25]. Mozart as a classical era composer is overlapping his predecessors but also being distinctive from them. The cluster of Beatles is loose than Nirvana, maybe reflecting that music style of Beatles is more diverse.

4.3 Music Genre Classification We perform one vs one music genre classification as a validation of the the power of `MusicHON` and proposed features. If the `MusicHON` representation and the features we proposed capture meaningful aspects of music, we should be able to use these features to predict music genre. For branching and repeateness, we take $\mathcal{F}_B^m(0.1)$ and $\mathcal{F}_{R|l=5}$. We denote this as `MusicHON` in the results.

We also consider graph properties of `MusicHON` as additional features. The graph properties we consider are number of nodes, number of edges, diameter, shortest path length, density, clustering coefficient and modularity. We denote this with `MusicHON+` in the results.

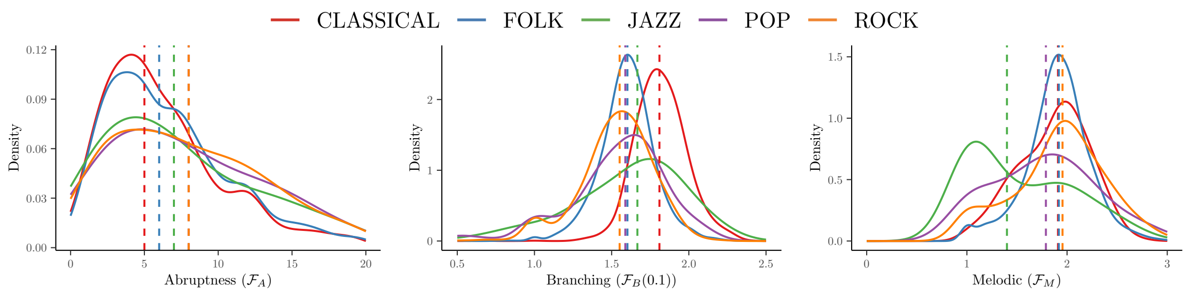
Baselines We compare the performance of `MusicHON` and `MusicHON+` against two baselines. Ferreti [9] represented music as a simple network, and described various features extracted from this representation that helps to describe different type of music. Therefore, to compare our `MusicHON` features, we also generated simple network representations of the music pieces and extracted the features described in [9]. We will refer this as `SimpleNetwork` in the subsequent discussion.

Out of the features we propose in Section 3.3, only branching and repeatedness are applicable to a simple network. Therefore we also perform experiments where we use these two features (calculated from the simple network) along with the `SimpleNetwork` features. We refer to this as `SimpleNetwork+` in the discussion that follows.

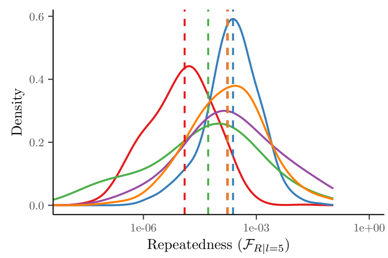
Experiment Setting Since the goal is to evaluate the goodness of the features, we use multiple classifiers: Support Vector Machine (SVM) [26], Random Forest (RF) [3] and Multilayer Perceptron Neural Network (MLP) [20]. For SVM, we use a radial basis function kernel; for RF, we use 100 trees; and for MLP, we use two hidden layers of sizes 10 and 5. To handle the issue of class imbalance, we undersample the majoritiy class to match the minority class.

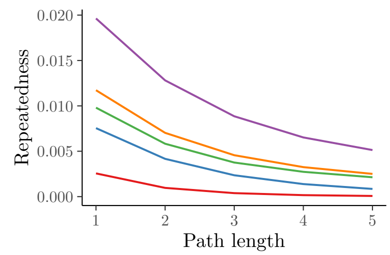
Classification Results Fig. 4(a) shows the results for the classification between Classical vs Folk, Classical

[2]The plots of the first three principal components are given in the supplementary material.

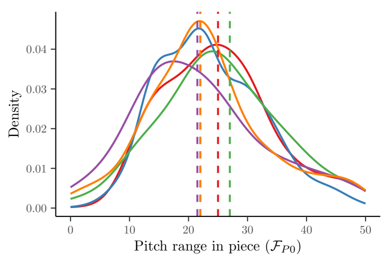


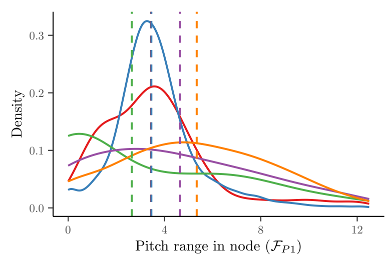
(a) Abruptness (b) Branching (c) Melodic

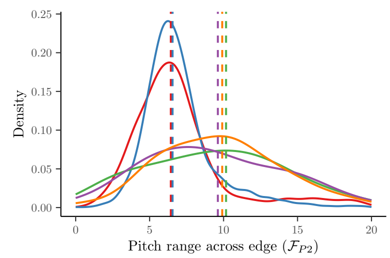




(d) Repeatedness (e) Repeatedness of different length





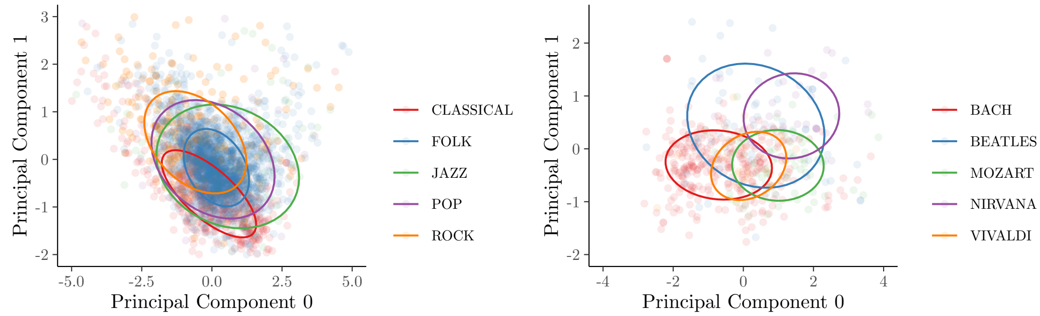


(f) Pitch range in piece (g) Pitch range in node (h) Pitch range across edge

Feature	Classical	Folk	Jazz	Pop	Rock
$\mathcal{F}_A$	5	6	7	8	8
$\mathcal{F}_B$	1.80	1.60	1.66	1.58	1.55
$\mathcal{F}_{R\|l=5}$	1.24×10^{-5}	2.40×10^{-4}	5.25×10^{-5}	1.69×10^{-4}	1.74×10^{-4}
$\mathcal{F}_M$	1.90	1.91	1.40	1.78	1.95
$\mathcal{F}_{P0}$	25	22	27	21	22
$\mathcal{F}_{P1}$	3.44	3.45	2.64	4.64	5.33
$\mathcal{F}_{P2}$	6.43	6.55	10.17	9.61	9.90

(i) Median value of features of different genres.

Figure 2: **The distributions of features generated from `MusicHON` for pieces of different genres**. The median of each distribution is shown as vertical lines and additionally reported in the table. The observations align with our common perception of music genres. Pop, Rock and Jazz are more "abrupt" than Folk and Classical. Classical music has significantly more "branches" than other genres. For melodic, we observe that Jazz is significantly less "melodic" than other genres. For repeatedness, we can see that Classical are less repetitive than other genres; repeatedness score also decreases with the increase of path length. For pitch ranges, we observe limited difference of music genres in pitch range in piece. However, for pitch range in node and pitch range across edge, we observe that Classical and Folk are significantly lower than other genres.



(a) Genres (b) Artists

Figure 3: **Principal components of the features for different genres and artists.** We observe interesting patterns such as: the centroid for classical music cluster is further away from other genres; Bach and Vivaldi are overlapped a lot possibly due to they are both Baroque era composers, etc.

vs Jazz and Classical vs Pop.[3] We report the Area under Curve (AUC) of the Receiver Operating Characteristic (ROC) curve for 5-fold cross validation for different classifiers (denoting on the x-axis) and features used (denoting with different colors).

As we can observe, Random Forest achieves higher accuracy comparing to SVM and MLP. Almost in all the cases, `MusicHON` and `MusicHon+` features outperforms `SimpleNetwork` and `SimpleNetwork+`. In most of the cases, the improvement is statistically significant. On comparing `MusicHON` and `MusicHON+` we can see that `MusicHON+` slightly improves the performance on `MusicHON`. With new features directly related to music, `SimpleNetwork+` slightly outperforms `SimpleNetwork`, but in many cases still underperform `MusicHON` and `MusicHON+`. This further demonstrates the usefulness of the proposed features and the power of higher-order dependencies.

We also perform classification between selected artists/composers. The results are provided in the supplementary material due to space limitations, and similar observations can also be made under those cases.

Feature Importance To investigate if the `MusicHON` features correspond to what we understand about music, we investigate the feature importance assigned by the Random Forest classifier. A feature is assigned a higher importance score if it can better separate out the different classes.

Fig. 4(b) shows the feature importance for Classical vs Folk, Classical vs Jazz and Classical vs Pop.[4] We can see that branching and repeatedness is important for differentiating Classical from Folk and Pop. This is expected since Classical has in general a more complex structure and less repetitive than Folk and Pop. The most import feature to distinguish Classical from Jazz is melodic and average pitch range in edge. This is possibly due to that Jazz has a lot of improvisations which makes it less melodic than classical and since it is a more modern music genre, the pitch ranges is different from Classical. These observations indicate that `MusicHON` features are related to characteristics of different music genres.

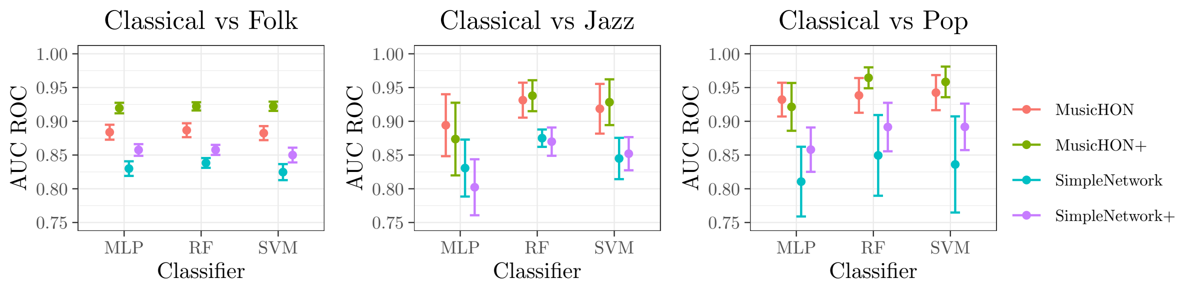
4.4 Case Study To further understand the relationship between a music piece and its `MusicHON` representation, we conduct a case study on the song *Blackbird* by *The Beatles* (see Fig. 5). To make the visualization easier to understand, we transform the relative MIDI coding back to the 12 notes in music.

From the network, we can clearly identify the opening and the end of the first section of the piece (marked with ellipses in Fig. 5). Interestingly, these two parts are connected together in the network and form a community, which may reflect the concept of "going home" in music. We also located the highest-order node in this network, the node "A—B.D.C" (marked with square), and found that it appears in the connection of two sections within the verse. The case study suggests that `MusicHON` can provide information on the higher-level structure of a music piece.

4.5 Running Time We also experimentally evaluate the time it takes to generate `MusicHON` against the trajectory length (length of the symbolic representation of the musical piece), and the running time to extract all the proposed features against number of edges in `MusicHON`(shown in Fig. 6). We can observe that for both `MusicHON` generation and feature extraction, the

[3] The results for other genre pairs are provided in the supplementary material.

[4] Other results are provided in supplementary material.



(a) AUC scores.

Genre 1	**Genre 2**	$\mathcal{F}_A$	$\mathcal{F}_B$	$\mathcal{F}_{R\|l=5}$	$\mathcal{F}_M$	$\mathcal{F}_{P0}$	$\mathcal{F}_{P1}$	$\mathcal{F}_{P2}$
Classical	Folk	0.07	**0.26**	**0.24**	0.09	0.10	0.11	0.09
Classical	Jazz	0.08	0.11	0.09	**0.21**	0.16	0.10	**0.21**
Classical	Pop	0.11	**0.20**	**0.17**	0.08	0.13	0.12	**0.17**

(b) Feature Importance.

Figure 4: **Genre classification results on Classical vs Folk, Jazz and Pop. (a) AUC scores.** The x-axis denotes different classifier used and the y-axis is the (mean ± standard deviation) of the AUC score from 5-fold cross validation. Both `MusicHON+` and `MusicHON` perform significantly better than `SimpleNetwork` and `SimpleNetwork+` in most of the cases. `MusicHON+` slightly outperforms `MusicHON`. **(b) Normalized feature importance from Random Forest classifier.** Higher values indicates that the feature is more important. The two most important features for each classification are indicated in bold. We observe that branching and repeatedness are more important distinguishing Classical from Folk and Pop, while melodic and pitch range across edge are more important for Classical vs Jazz.

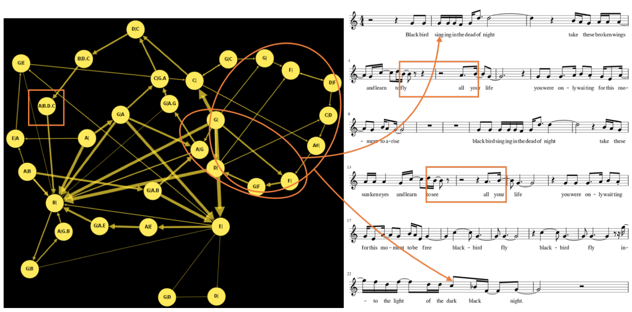
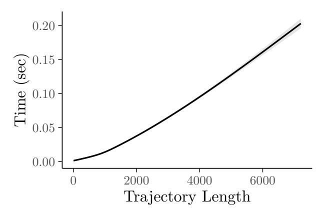
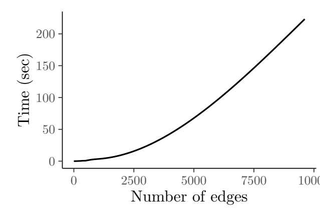


Figure 5: **Case Study of *Blackbird* by *The Beatles* using `MusicHON` representation.** From the network, we can identify the opening and the end of the first section of the piece. We also spot that the highest-order node "A—B.D.C" appears in the connection of two sections within the verse.



(a) Running time to generate `MusicHON` against trajectory length.



(b) Running time to extract the proposed features against the number of edges in `MusicHON`.

Figure 6: **Running time to generate `MusicHON` and feature extraction.**

growth of running time is close to linear, showing that our method is very scalable.

5 Conclusion and Future Work

In this paper, we propose modeling music pieces using `MusicHON` which can capture higher-order dependencies in music pieces. We propose several meaningful features extracted from this representation.

Feature distribution and PCA analysis shows that the features from `MusicHON` can reflect the differences between music genres and artists. Experiments show that using `MusicHON` features for music genre classification outperforms the ones based on simple networks. Feature importance analysis shows that important features to distinguish genres align with common music perception. Case study shows that `MusicHON` can provide information on the higher-level structure of a music piece. The above results demonstrate the power of higher order networks in understanding music.

In this work, we consider only one track in the music piece and do not consider the duration of the notes. In music, the interaction between different tracks and duration of the notes are very important. We leave them as future work for further improvement of our model.

6 Acknowledgments

We thank Josefine Bohr Brask from University of Exeter and Carlos Marcelo from University of Chicago for providing ideas and suggestions on this project.

References

[1] M. G. Armentano, W. A. De Noni, and H. F. Cardoso, *Genre classification of symbolic pieces of music*, Journal of Intelligent Information Systems, 48 (2017), pp. 579–599.

[2] S. Blackburn, D. De Roure, et al., *A tool for content based navigation of music.*, in ACM multimedia, vol. 98, 1998, pp. 361–368.

[3] L. Breiman, *Random forests*, Machine learning, 45 (2001), pp. 5–32.

[4] Z. Cataltepe, Y. Yaslan, and A. Sonmez, *Music genre classification using midi and audio features*, EURASIP Journal on Advances in Signal Processing, 2007 (2007), p. 036409.

[5] D. C. Corrêa, A. L. Levada, and L. d. F. Costa, *Finding community structure in music genres networks.*, Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), (2011), pp. 447–452.

[6] M. S. Cuthbert and C. Ariza, *music21: A toolkit for computer-aided musicology and symbolic music data*, Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010), (2010), pp. 637–642.

[7] R. B. Dannenberg, *Music representation issues, techniques, and systems*, Computer Music Journal, 17 (1993), pp. 20–30.

[8] A. De Montis, M. Barthélemy, A. Chessa, and A. Vespignani, *The structure of interurban traffic: a weighted network analysis*, Environment and Planning B: Planning and Design, 34 (2007), pp. 905–924.

[9] S. Ferretti, *On the complex network structure of musical pieces: analysis of some use cases from different music genres*, Multimedia Tools and Applications, 77 (2018), pp. 16003–16029.

[10] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith, *Query by humming*, Readings in multimedia computing and networking, (2002), p. 216.

[11] M. F. Heath, M. C. Vernon, and C. R. Webb, *Construction of networks with intrinsic temporal structure from uk cattle movement data*, BMC Veterinary Research, 4 (2008), p. 11.

[12] T. Järvinen, P. Toiviainen, and J. Louhivuori, *Classification and categorization of musical styles with statistical analysis and self-organizing maps*, in Proceedings of the AISB Symposium on Musical Creativity, 1999, pp. 54–57.

[13] I. Joliffe and B. Morgan, *Principal component analysis and exploratory factor analysis*, Statistical methods in medical research, 1 (1992), pp. 69–95.

[14] P. Kaluza, A. Kölzsch, M. T. Gastner, and B. Blasius, *The complex network of global cargo ship movements*, Journal of the Royal Society Interface, 7 (2010), pp. 1093–1103.

[15] A. Kotsifakos, E. E. Kotsifakos, P. Papapetrou, and V. Athitsos, *Genre classification of symbolic music with smbgt*, in Proceedings of the 6th international conference on PErvasive technologies related to assistive environments, ACM, 2013, p. 44.

[16] J. Lee, M. Lee, D. Jang, and K. Yoon, *Korean traditional music genre classification using sample and midi phrases.*, KSII Transactions on Internet & Information Systems, 12 (2018).

[17] J. McCormack, *Grammar based music composition*, Complex systems, 96 (1996), pp. 321–336.

[18] M. E. Newman, *Analysis of weighted networks*, Physical review E, 70 (2004), p. 056131.

[19] M. Rosvall, A. V. Esquivel, A. Lancichinetti, J. D. West, and R. Lambiotte, *Memory in network flows and its effects on spreading dynamics and community detection*, Nature communications, 5 (2014), p. 4630.

[20] D. E. Rumelhart, G. E. Hinton, R. J. Williams, et al., *Learning representations by back-propagating errors*, Cognitive modeling, 5 (1988), p. 1.

[21] M. T. Schaub, J. Lehmann, S. N. Yaliraki, and M. Barahona, *Structure of complex networks: Quantifying edge-to-edge relations by failure-induced flow redistribution*, Network Science, 2 (2014), pp. 66–89.

[22] I. Scholtes, N. Wider, R. Pfitzner, A. Garas, C. J. Tessone, and F. Schweitzer, *Causality-driven slow-down and speed-up of diffusion in non-markovian temporal networks*, Nature communications, 5 (2014), p. 5024.

[23] J. Serrà, Á. Corral, M. Boguñá, M. Haro, and J. L. Arcos, *Measuring the evolution of contemporary western popular music*, Scientific reports, 2 (2012), p. 521.

[24] A. Swift, *A brief introduction to midi*, URL http://www. doc. ic. ac. uk/˜nd/surprise_97/journal/vol1/aps2, 6 (1997).

[25] M. Talbot, *Vivaldi*, Routledge, 2017.

[26] V. Vapnik, *The nature of statistical learning theory*, Springer science & business media, 2013.

[27] H. Vinet, *The representation levels of music information*, in International Symposium on Computer Music Modeling and Retrieval, Springer, 2003, pp. 193–209.

[28] J. Xu, T. L. Wickramarathne, and N. V. Chawla, *Representing higher-order dependencies in networks*, Science advances, 2 (2016), p. e1600028.