Contents lists available at ScienceDirect

Visual Informatics

journal homepage: www.elsevier.com/locate/visinf

Visual exploration of latent space for traditional Chinese music

ARTICLE INFO

Article history:
Received 1 Feb 2020
Received in final form X XX 2020
Accepted X XX 2020

Keywords: Music information retrieval, Latent space analysis, Long Short-Term Memory, Autoencoder, Traditional Chinese music

ABSTRACT

Generating compact and effective numerical representations of data is a fundamental step for many machine learning tasks. Traditionally, handcrafted features are used but as deep learning starts to show its potential, using deep learning models to extract compact representations becomes a new trend. Among them, adopting vectors from the model's latent space is the most popular. There are several studies focused on visual analysis of latent space in NLP and computer vision. However, relatively little work has been done for music information retrieval (MIR) especially incorporating visualization. To bridge this gap, we propose a visual analysis system utilizing Autoencoders to facilitate analysis and exploration of traditional Chinese music. Due to the lack of proper traditional Chinese music data, we construct a labeled dataset from a collection of pre-recorded audios and then convert them into spectrograms. Our system takes music features learned from two deep learning models (a fully-connected Autoencoder and a Long Short-Term Memory (LSTM) Autoencoder) as input. Through interactive selection, similarity calculation, clustering and listening, we show that the latent representations of the encoded data allow our system to identify essential music elements, which lay the foundation for further analysis and retrieval of Chinese music in the future.

1. Introduction

As an active research branch of Information Retrieval, Music Information Retrieval (MIR) is a multidisciplinary research field aiming to create effective representations for digital music, analyze salient music information, develop content-based retrieval schemes, and provide user-friendly interfaces (e.g., Downie, 2004). Specifically, there are two main directions in MIR: extracting features from raw audios to describe music, and utilizing additional factors such as lyrics, performers, and listener preferences to form content-rich music descriptors. We are interested in the former in this paper. Typically, the related MIR tasks include feature extraction (e.g., Laden and Keefe (1989); Eck and Schmidhuber (2002); Nam et al. (2012); Dieleman and Schrauwen (2013); Dieleman and Schrauwen (2014); Costa et al. (2017)), similarity comparison (e.g.,West and Lamere (2006); Slaney et al. (2008); Schluter and Osendorfer (2011); Janssen et al. (2017)), classification (e.g., Costa et al. (2017); Yu et al. (2020)), and applications such as music recommendation (e.g.,McFee et al. (2011); van den Oord et al. (2013); Koenigstein et al. (2011)), music generation (e.g., Kim et al. (2009); van den Oord et al. (2016)), music transcription (e.g., Sigtia et al. (2015)), music visualization (e.g., Foote (1999); Cooper et al. (2006); Yim et al. (2009)) and other inspiring directions. Among them, developing compact and effective music representations is a critical step of the process.

In traditional MIR, music information is not always well preserved, and the feature engineering process involved is tedious and time-consuming. A turning point in MIR is the recent development of machine learning and deep learning techniques. Inspired by computer vision and natural language processing (NLP), scientists try to extract audio features using Convolutional Neural Networks (CNNs) (e.g., Dieleman and Schrauwen (2014); Lee et al. (2017); van den Oord et al. (2016); Choi et al. (2016); Korzeniowski and Widmer (2016)), Recurrent Neural Networks (RNNs) (e.g., Sigtia et al. (2015); Li et al. (2016)) and Autoencoders (e.g., Meyer et al. (2017)). Based on this, they perform music classification, manipulation, prediction, and synthesis.

Despite the remarkable performance, features learned by deep neural network models are difficult to explain due to the black-box nature of the models. To this end, many researchers attempt to utilize visualization methods to interpret

deep neural networks as well as the features / embeddings learned by these models in computer vision and NLP domains (e.g., Zhu and Chen (2007); Zeiler and Fergus (2013); Garcia-Gasulla et al. (2015); Liu et al. (2018); Camargo and González (2014a)). Some studies attempt to visualize music features such as MFCCs (e.g.,Panda et al. (2019)), but not much visual analysis has been done to understand music features extracted from deep neural networks.

Besides, it's noticeable that a majority of MIR tasks are focused on western music, especially pop music and western classical. Traditional Chinese music, as a rich and complex art form, is almost neglected. There are two factors contributing to this phenomenon. First, traditional Chinese music is known by a relative smaller group compared to western music, which leads to a relatively smaller collection of datasets. Second, the most influential MIR conference, Music Information Retrieval Evaluation eXchange (MIREX) has tasks where the datasets are western music by a large majority. In addition, instead of being composed by harmonies, traditional Chinese music is more about the melodies. Such inherited differences between western music and traditional Chinese music make it difficult for researchers to directly apply methodologies developed for western music analysis to traditional Chinese music (e.g., analyzing features extracted from deep neural networks).

Therefore, our work aims to contribute to this neglected area, information retrieval for traditional Chinese music. Specifically, we want to provide a tool that can help retrieve information from traditional Chinese music, compare performances to assist teaching and learning, explore properties of different music groups, and generate a smooth blending between different music pieces. To achieve these goals, the fundamental step is music representation analysis and evaluation which is the focus of our paper.

To begin our studies, a proper dataset is required. For this we first built a traditional Chinese music collection from over 30 albums recommended by domain experts, and then we transform the audios in the collection to the spectrograms with labels of instrument and performer for each music. Spectrograms capture the frequencies and amplitudes from raw audio wave signals and hence provide a foundation for further music analysis. Due to the complexity and high dimensional nature of the spectrograms, however, we cannot directly apply visual analysis to them. To resolve this issue, we utilize two deep neural networks, a fully-connected Autoencoder and a Long Short-Term Memory (LSTM) Autoencoder, to extract compact music representations. Taking spectrograms column by column as input, the fully-connected Autoencoder can learn a compressed latent representation for each column which will be treated as the note latent vector. Since music consists of time-varying signals, features in the time domain containing essential temporal information. Thus, we feed the note latent vectors resulted from one music segment with a proper sampling rate into a LSTM Autoencoder, from which we get a latent vector representing the time-dependent signal, called the segment latent vector.

In an attempt to offer visual exploration and analysis of the latent representations, we propose a visual analytics system, MusicLatentVIS. The system is composed of a two-dimensional (2D) projection of the latent space, a list of visualizations including heatmaps, parallel coordinates, and a menu for users to select groups of music or performers. Users can click on the 2D projection and listen to the specific segment of the music to explore the latent space. By moving around in the 2D projection, discovering clusters and comparing latent vectors, users can locate some interesting properties of music. Additionally, we use several evaluation tasks to demonstrate the usefulness of our system.

2. Related work

2.1. Feature Extraction in Music Information Retrieval

In MIR field, there has long recognized demand for effective and informative music representations instead of raw audios. As stated by Peeters (2004), traditional MIR uses handcrafted features to represent audios, such as the autocorrelation coefficient, zero-crossing rate (e.g., Kumarbanchhor and Khan (2012); Changsheng Xu et al. (2003)), inharmonicity (e.g., Agostini et al. (2003)), spectral centroid (e.g., Agostini et al. (2003)), spectral contrast (e.g., Jiang et al. (2002)), Mel-frequency cepstral coefficients (MFCCs), short-time Fourier transform (STFT)) and so on. However, employing these handcrafted features always requires a pre-processing step such as dimensionality reduction (e.g., Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA)) to avoid the curse of dimensionality and remove redundant features.

Motivated by the recent success of utilizing deep learning models in computer vision and NLP, some MIR studies take advantage of neural networks for feature extraction to reduce the feature engineering efforts. Dieleman and Schrauwen (2014) applied 1D CNN directly to raw audio signals for feature learning. Lee et al. (2017) proposed a sample-level Deep CNN model with smaller filter length and subsampling length to learn representations from waveforms. van den Oord et al. (2016) proposed WaveNet, a generative model utilizing dilated convolution to extract features and generate raw audio waveforms autoregressively. Li et al. (2016) adopted Deep Bidirectional Long Short-Term Memory (DBLSTM) to capture context correlation of music feature sequence. Chuan et al. (2018) explored word2vec, a shallow neural network proposed by Mikolov et al. (2013) which can represent words by vectors with semantic meanings, as a feature extraction method to capture meaningful relationships in music.

2.2. LSTM Autoencoder for Feature Extraction

Since LSTM has the ability to model temporal dependencies of a sequence and Autoencoder works great in feature extraction, the combination of them which is called a LSTM Autoencoder, is widely used in feature extraction of time-varying data. First proposed by Srivastava et al. (2015), LSTM Autoencoder was used to learn video representations. In addition, Marchi et al. (2015) utilized a denoising Autoencoder with bidirectional LSTM to process auditory spectral features. Zhao et al. (2018) proposed a robust LSTM Autoencoder for effective face de-occlusion. Their LSTM Autoencoder consists of two parts, one is face encoding and another is occlusion removal. Besides

videos, LSTM Autoencoder also demonstrates its potential to model audios. For instance, Tang et al. (2018) used a LSTM Autoencoder to learn the mapping between acoustic and motion features for dance synthesis.

2.3. Visual Analysis of Latent Space

Autoencoder is a neural network designed for extracting latent representations of datasets; however, since the Autoencoder's encoding schema is unclear, the latent vectors extracted by Autoencoders are not guaranteed to be human-interpretable as stated by Hristov et al. (2018) and Liu et al. (2019). To interpret the latent space, many visual methods are proposed. First, researchers have to verify if the latent space obtained from Autoencoder is trustworthy, e.g., if the internal correlations among data samples are preserved in the latent space. The widely used method is projecting high dimensional data into 2D space by employing t-Distributed Stochastic Neighbor Embedding (tSNE), Principal component analysis (PCA), or Uniform Manifold Approximation and Projection (UMAP). By comparing the clusters in the original and latent spaces, people can observe if the properties of the original data are preserved in the latent vectors. Nabney et al. (2005) provided interactive clustering for recognizing the collection of documents with similar topics. Second, domain specific methods are adopted to visualize semantic meanings in latent space. Liu et al. (2019) created a map of attribute vectors for opposing concepts. Ji et al. (2019) clustered latent variables according to hierarchical structures in medical documents. Camargo and González (2014b) projected both images and text on the latent space to demonstrate the relationship between them.

2.4. Visual Analysis of Music Representations

Previously, researchers have been working on the design of proper visual representations of music to improve music understanding, e.g., Cruz et al. (2018) proposed to use shapes, colors and movements to represent music structure. Inspired by the recent success on visualizing embeddings in NLP, Panda et al. (2019) extracted features of music segments from probabilistic topic model's latent space, then the resulting probabilistic labels are used to interpret music. For visualization, they chose a doughnut chart to represent probability distribution and a line graph to visualize distributions over time. But the actual underlying semantic meaning of latent representations is missing.

3. Music latent feature extraction

As we mentioned before, the fundamental goal of our work is visual analysis and evaluation of music representations. This section mainly focuses on the process of extracting music features. First we choose the spectrogram as the initial representation of music since spectrograms contain both frequency and amplitude information of the music signals. But as there are redundancy in the spectrograms, we employ two neural network models to extract more compact features. Specifically, we propose to use latent space of Autoencoders to generate our music features, namely a fully-connected Autoencoder for note latent vectors and a Long Short-Term Memory (LSTM) Autoencoder for music segment latent vectors. After that, we can perform visual analysis in the music latent space. The workflow of our music feature generation process is illustrated in Figure 1(A):

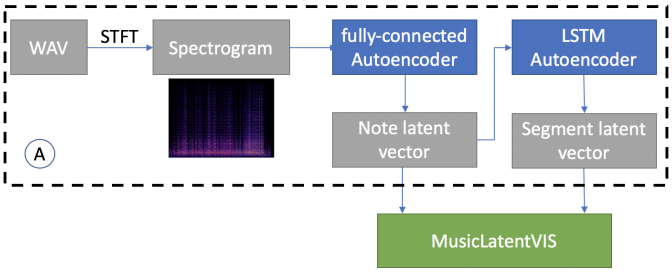

Fig. 1. Overview

3.1. Data Preprocessing

To get a comprehensive dataset of traditional Chinese music, we built a dataset that consists of 373 music pieces in the spectrogram format, covering 17 instruments performed by more than 60 musicians. Most of the music are collected from commercial CDs. There are three major reasons for us to make use of commercial CD collections instead of merely recordings from musicians whom we have access to. First, the quality of the commercial CDs in general is better due to the competitive market. Second, professional music studios have better recording equipments. Third, the cost would be too high if we record data as large as this collection by ourselves.

From the technical points of view, music can be seen as a sequence of musical notes or note combinations (so called chords in western music), and musical notes can be decomposed into certain frequencies. For this reason, we compute spectrograms (a frequency-time representation generated by the short-time Fourier transform as stated by Muller et al. (2011)) from the raw audios instead of using waveforms (an amplitude-time representation) as our Autoencoder input.

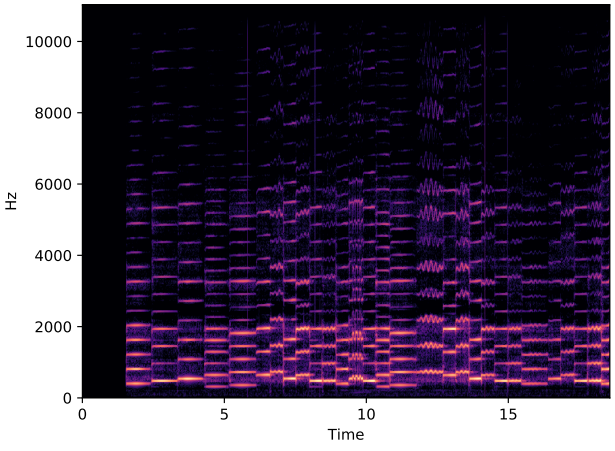

Fig. 2. A spectrogram segment of Erhu music, Liang Xiao. The column has 501 frequency bins, each time step is 0.025s, and the color represents the value of amplitude. The brigher the color, the bigger the amplitude.

A spectrogram is a 2-dimensional matrix, whose column represents frequency bins, row represents time, and the values in the matrix are the amplitudes of certain frequency bins at certain time steps. From the spectrogram in Figure 2, we notice that each musical note is a complex tone composed by a significant fundamental frequency (denoted as f_0) and overtones. According to Wood and Bowsher (1980), overtones consist of harmonic partials (multiples of f_0, such as $2f_0$, $3f_0$, $4f_0$, ...) and inharmonic partials. As the study by Wood and Bowsher (1980), the fundamental frequency determines the pitch, the overtones correspond to the timbre of instruments and the amplitudes contributes to the loudness of the notes.

Although the spectrogram is a useful representation, the matrix nature of it implies it is high dimensional, and is often sparse with inherent redundancy which is not ideal for visual analysis. As we mentioned before, a musical note is a combination of f_0 and overtones, so at least 2/3 frequency bins are not necessary to represent a note. However, music spectrograms can be further grouped into several frequency bands with different contributions, which means a more concise representation is possible.

Therefore, the next step is to extract more compact information from the spectrograms, for which we use deep neural networks. We notice that the distribution of amplitudes in the frequency bins is not in an ideal scale for neural networks, so before we feed data into our neural networks, we convert the original data into the log scale and then perform min-max normalization on the log scale data.

3.2. Model Architecture

In automatic feature engineering, Autoencoder, an unsupervised deep learning model, is widely used to extract compact representations for its input. Consisting of an encoder and a decoder, an Autoencoder can encode the input x into a latent representation $f(x)$ using its encoder f. Then the latent representation $f(x)$ can be decoded as the output $g(f(x))$ through decoder g. The goal of Autoencoder is to minimize the difference between the input x and the reconstructed output $g(f(x))$.

For dimensionality reduction purposes, Autoencoders are often designed with a constraint that the latent representation $f(x)$ has a smaller dimension than the input x. In this way, Autoencoders are forced to learn the most essential features present in the data. As a result, the learned latent representation $f(x)$ can be utilized as a compact feature of the input.

Since the spectrogram's two axes have different meanings, one is time and the other is frequency, we can introduce two kinds of features (i.e. with and without temporal correlation). In our work, we utilize two variations of Autoencoders for feature extraction.

3.2.1. Fully-connected Autoencoder

As stated before, due to the redundancy of spectrograms along the frequency axis, it is feasible to use an Autoencoder to extract a note feature from each column of the spectrogram without temporal information. Since people can recognize music information such as pitch, loudness and timbre from a single musical note, no matter how short it is, one column of a spectrogram is enough to describe a musical note except its duration. For this reason, we also call one column of the spectrogram as a note sample.

We choose an Autoencoder with fully-connected layers (fully-connected Autoencoder) to learn note features. This Autoencoder includes one input layer and two hidden layers that are fully-connected as an encoder, and another two fully-connected layers with activation as a decoder. We use Mean Squared Error (MSE) as the loss function to measure the difference between the input and output and force the model to learn features from the input.

The encoder encrypts a column of the input spectrogram at the dimensions of 501×1 to a latent vector of size 32×1, and the decoder decrypts the latent vector back to a 501×1 vector. Through careful experiments, we design the structure of our model as shown in Figure 3.

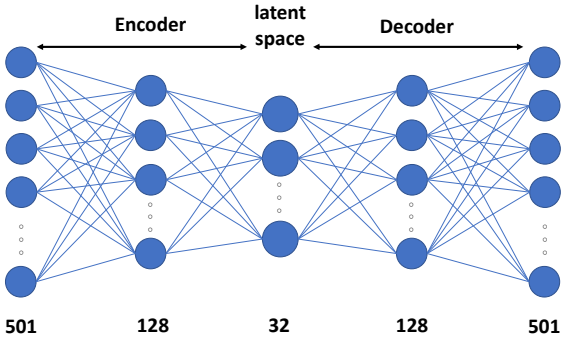

Fig. 3. Fully-connected Autoencoder.

After training, the encoder part of the fully-connected Autoencoder is used to generate compact column-wise features (we call that note latent vectors) of input spectrograms. The evaluation of latent vectors in Section 4.2 demonstrates that note latent vectors can successfully capture the three features of a music note: timbre, pitch, and loudness.

3.2.2. LSTM Autoencoder

Our goal of this section is to extract latent vectors from a duration of music, i.e., a music segment.

Although convolution performs well when extracting features, they are not suitable for modeling temporal information for time-series data such as music. Recurrent neural networks (RNNs) on the other hand, especially LSTMs, are capable of learning the temporal correlation of input sequences. Generally, a LSTM consists of a *cell state* which is the memory of the previous input, a *forget gate* to remove information from memory, an *input gate* to regulate what information to add into memory, and an *output gate* to control what to output based on the input and memory. This scheme allows LSTMs to extract both long-term and short-term dependencies of input sequences.

To extract music features with temporal correlation over time, we adopt an LSTM Autoencoder. As a variation of the regular Autoencoder, a LSTM Autoencoder also consists of an encoder and a decoder. Inspired by the sequence-to-sequence

model proposed by Sutskever et al. (2014), which is often used to translate a sentence to another language, the LSTM Autoencoder can be similarly trained to convert a long sequence into a short representation, and then convert it back to the original size. Specifically, the LSTM encoder processes an input sequence step by step, after feeding in the whole sequence, the encoder's output is taken as the compressed latent representation. The LSTM decoder then decodes the latent representation step by step and outputs a reconstructed sequence.

Before we build our LSTM Autoencoder model, we need to decide how to represent a music segment as a sequence.To represent a single music segment, we can use a spectrogram of T time steps with each time step has 501 elements, or we can take a set of note latent vectors sampled from this segment.

Advised by our domain experts, who are long-time Chinese music performers, we know in order to differentiate or compare the semantic meaning of music segments, the duration of a music segment needs to be at least 10 seconds. Due to the fact that the sampling rate for the original spectrogram is 20kHz which means one column in spectrogram is only a small slice at a duration of 0.025 seconds, a spectrogram slice of 10 seconds will have 400 time steps (the length on time axis) which is difficult for a typical LSTM to iteratively process through time. Nevertheless, as stated in Section 3.1, each instrument has a different combination of dominant frequencies which means the spectrogram has inherent redundancy along the frequency axis.

In fact, to capture both the melody and the tempo of the music, our goal is not to only extract the music notes. For this reason, a uniform sampling along the time axis is more desired. Considering the meaningful duration of music segments, redundancy of the spectrograms, and the computational complexity and reconstruction accuracy, after extensive experiments, we chose to use a downsampled representation by dividing each spectrogram into bins of size 20 along the time axis and picking the first time step from each bin. This sampling rate can reduce the redundancy and preserve the music melody. As for the redundancy along the frequency axis, instead of manually determining the dominant frequencies via simple dimensionality reduction schemes, we send columns of the downsampled spectrogram to the trained fully-connected Autoencoder to get compact note latent vectors (each has 32 dimensions). In other words, we use a set of 20 note latent vectors to represent a 10-second music segment. One limitation here is that we are making a trade-off between accuracy and computational complexity. One possible future work is to use more adaptive downsampling methods to reduce redundancy.

These 20 note latent vectors are time-dependent, for which the LSTM model needs to process them step by step. To facilitate a more effective training, we further reshape the 20 note latent vectors into 4 time steps with each time step consisting of 5 concatenated vectors. As a result, the final input for the LSTM Autoencoder is a time-varying sequence with 4 time steps. Each time step is a 160-dimensional vector ($32 \times 5 = 160$). In this way, we can consider both the temporal and spacial correlation of music segments.

The architecture of our LSTM Autoencoder is a stacked LSTM as shown in Figure 4. Figure 4(A) and Figure 4(B) are the encoder and decoder respectively. As discussed before, we have 4 time steps in total. At each time step during the encoding process, we feed a vector of 160 dimensions into the encoder and update the cell state which incorporates the information of current time step. Then we get an output vector of size 16 of this time step. The output of the encoder's last time step is taken as the latent representation of this music segment. We named this vector as segment latent vector. During the decoding process, the segment latent vector is given as an input for each time step of the decoder. The loss function is Mean Squared Error (MSE). Once the model is well trained, given a collection of note latent vectors representing a music segment, the encoder part can extract the corresponding segment latent vector.

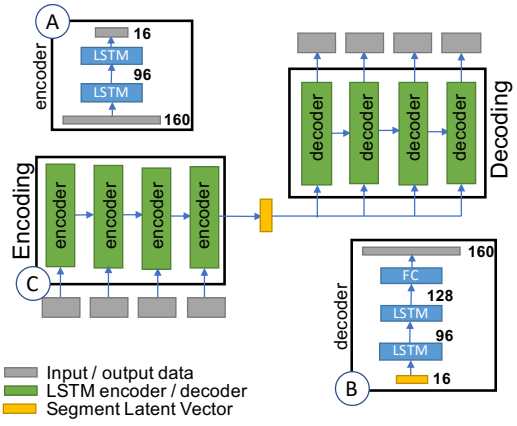

Fig. 4. LSTM Autoencoder. (A) and (B) are architectures of the encoder and decoder respectively. In (C), each green encoder block represents a time step during encoding phase and gray blocks represent input data for each time step. After encoding, a segment latent vector (the yellow block) is generated.

In summary, LSTM Autoencoder has two advantages: (1) like regular Autoencoder, it performs as an unsupervised learning method for music segment feature extraction, e.g., similar inputs will have similar compressed representations. (2) it is capable to learn complex temporal information.

4. MusicLatentVIS

To get better understanding of music data and facilitate the analytic process, we worked closely with domain experts E1 and E2. E1 is the conductor of university traditional Chinese music orchestra who has also been a Bamboo Flute performer for over 20 years. E2 is a traditional Chinese music performer who has played Zheng for 14 years and played Erhu for 12 years. They are both interested in studying music representations.

To show our visual interface and conduct experiments, we selected 324 music pieces played by four interesting instruments (i.e. Bamboo Flute, Erhu, Pipa and Zheng), since they are the representative traditional Chinese music instruments and attract the major audience. After fully-connected Autoencoder and LSTM Autoencoder are trained, we utilize them to produce two kinds of latent features: note latent vectors (column-wise spectrogram features without temporal information) and segment

latent vectors (with temporal information). Our work aims to retrieve information from the traditional Chinese music. More specifically, our goals are to compare, evaluate and analyze music latent representations through our interactive visual analytic system, MusicLatentVIS.

4.1. Visual Interface

Motivated by the recent success in feature extraction through neural networks in MIR, we are interested in the analysis and evaluation for music representations, especially for traditional Chinese music. To achieve this, we need to get an overview of all latent vectors. However, it is difficult to directly visualize high-dimensional vectors, so we apply t-SNE (t-distributed Stochastic Neighbor Embedding), a non-linear dimensionality reduction method. Due to the fact that t-SNE utilizes distributions of distances to model similarities and tries to minimize the difference between low dimensional objects and high dimensional objects, the visualization of 2D projection may show some important high dimensional structures from the data. Thus, t-SNE projection can assist the exploration of latent space (e.g., possible clusters in high dimensional latent space).

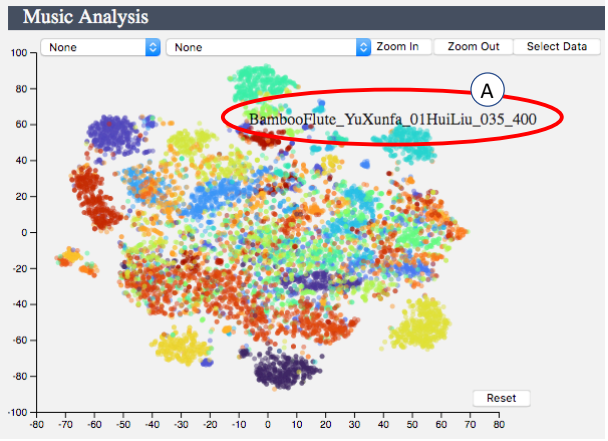

Fig. 5. t-SNE projection of music segment latent vectors. Each point, colored by performers, represents a music segment of ten seconds. We can see there are clusters of the same performers.

For instance, Figure 5 shows a t-SNE projection of music segment latent vectors, colored by performers. The coloring of data points in t-SNE can be modified later by selection and clustering. When users hover over a data point, related information including the instrument, the performer, the music name, and the start time index of this segment will appear as shown in Figure 5(A). To provide a clear view of those 2D points that are obstructed, our system allows users to brush an area or use mouse wheel to zoom in an projection area. Figure 6(B) is the new projection view after the user zooms in by dragging the mouse wheel. Advised by domain experts, to verify users' observations or hypothesis, our system also allows them to click on a single data sample in t-SNE projection and listen to that music segment.

Figure 7 shows a comparison of the t-SNE projection between spectrogram data and our note latent variables. As we mentioned in Section 3.2.2, the encoder of the trained fully-connected Autoencoder can extract compact features, undesired and redundant information such as noises and insignificant frequencies can be removed. The view in Figure 7 help identify the performance of the encoder. We project 501-dimensional note samples on the left panel and 32-dimensional note latent vectors on the right panel. Instruments (timbre information) are encoded by colors. Pitch clusters are labeled by texts, e.g., La4 denotes La in solfege at ocave4. Each note sample also has a hover over label containing pitch, loudness and timbre information in details.

To facilitate more effective browsing and querying, instead of considering all the data samples, it makes more sense if the user can focus on a subset of the data to perform comparison or information retrieval. To help analysis on a subset of data samples, MusicLatentVIS supports two kinds of selection methods on t-SNE projection: (1) users can select a subset of data by performer or by music name through the corresponding dropdown menu, (2) users can select data points by mouse brush (i.e., user can brush an area with mouse on t-SNE projection, points inside the rectangle brush area will be selected). After a selection, as shown in Figure 6(A), the selected data samples will be highlighted and kept for further analysis while the unselected ones will be grayed out.

The basic task for visual analysis is to understand the value distribution of latent representations. A parallel coordinates plot and a latent heatmap are generated for selected music segments. Parallel coordinates plots are useful for comparing multiple variables for a collection of music representations. In the parallel coordinates plot, each vertical axis represents one latent dimension and a polyline represents a music segment latent vector. Since in the visual encoding, change of positions is more effective than change of colors, parallel coordinates plots can easily find data patterns, e.g. the value distribution of music representations along one latent dimension. The color of each polyline shown in Figure 6(E) always agrees with the color of the corresponding point in Figure 6(A). Figure 6(C) is the heatmap for latent values where each row represents a segment latent vector and each column represents one latent dimension (i.e., suppose there are N selected segment latent vectors and each vector has k variables, the generated heatmap for these latent vectors will have N rows and k columns). In order to see the value distribution inside each dimension clearly, each dimension of the heatmap is colored separately. From the lowest to the highest value among all latent vectors at a specific dimension, the colors for the corresponding rectangles in this heatmap are set from blue to red. Unlike parallel coordinates plots which are better at tracing accumulated patterns but easier to have visual clusters, heatmaps are a type of space-filling visual representations which are are more effective in finding two-dimensional trends.

In order to identify which dimension contributes the most in differentiating segments in the selected subset, we calculate the variance of each dimension among all data samples in this subset. This gives us a clue to interpret dimensions. The columns (or coordinates axes) of both the latent heatmap and the parallel coordinates plot are sorted in an descending order by the variance of dimensions, meaning dimensions on the left will have higher variance and will contribute more on differentiating mu-

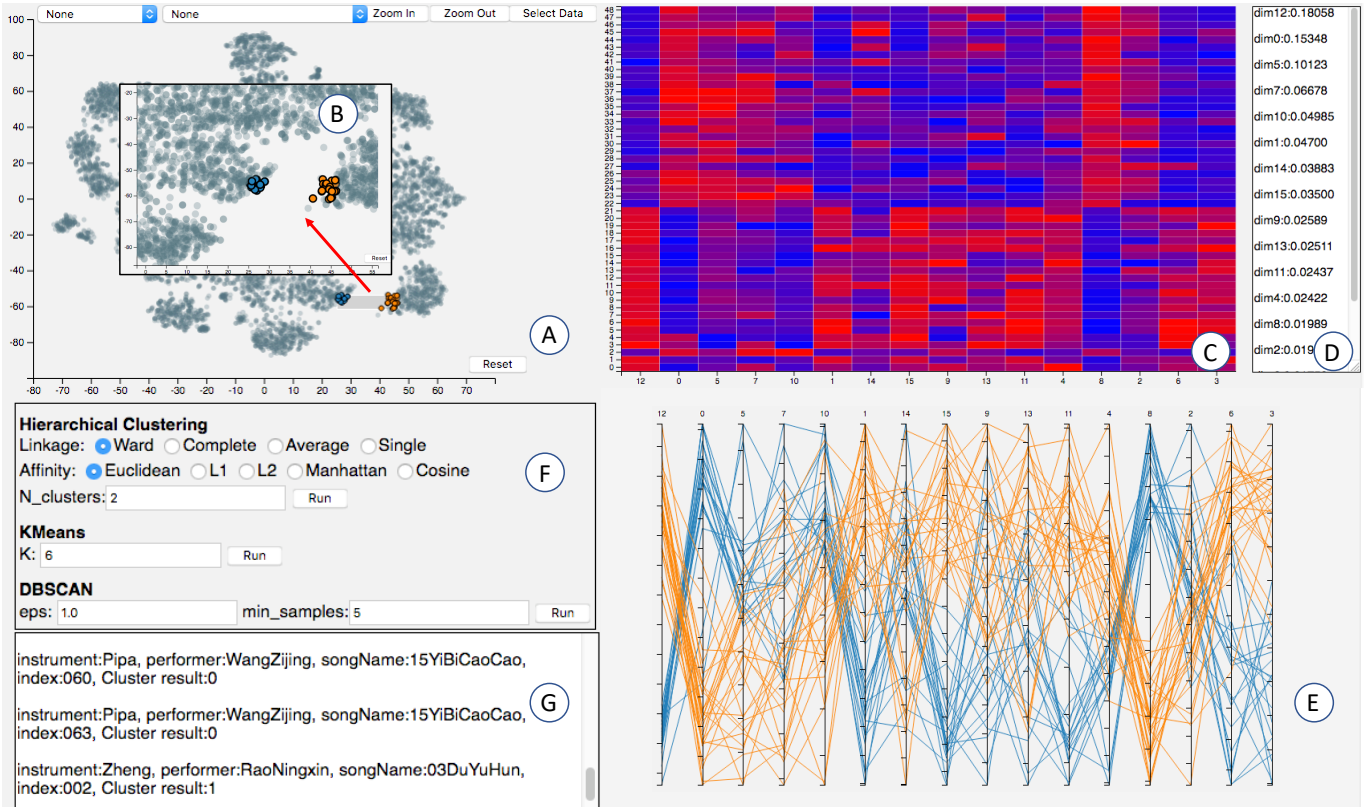

Fig. 6. An overview of our interactive visual analytic system. Details are in Section 4.1.

sic segments. Figure 6(D) is the text area showing the sorted variance along with the dimension index. In the example, we can see that the top 3 dimensions are dim-12, dim-0 and dim-5 with variance 0.18058, 0.15348 and 0.10123 respectively. They contribute the most on differentiating the blue and orange clusters.

To evaluate if the latent vectors can capture the similarity of the original music segments, our system allows users to generate heatmaps based on the cosine similarities among the selected segments, as shown in Figure 9. First the cosine distance between all pairs of the selected segment latent vectors are calculated. The color at position (i, j) in the similarity heatmap encodes the cosine similarity value between the i-th and the j-th latent vectors. A brighter color on the similarity heatmap indicates a higher cosine similarity which means more similar.

Also, since clustering can re-organize data into groups based on their similarities, it is reasonable to employ clustering methods on our music segment latent vectors for music information retrieval or comparison among music segments. To achieve this, our system allows users to employ a clustering technique as shown in Figure 6(F). The clustering results are calculated based on the n-dimension ($n = 16$) segment latent vectors from LSTM Autoencoder. According to users' needs or exploration interests, they can choose from Hierarchical Clustering, K-means and DBSCAN with user-defined settings. Figure 6(G) is the text area where the clustering result will appear.

4.2. Latent Feature Performance Evaluation

In this section we study whether the extracted features (i.e., note latent vectors and segment latent vectors) from our neural networks are meaningful and effective to represent the salient information of each note sample or each music segment of the spectrograms.

First, we generate music features from the two models in Section 3.2 after the training completed.

To compare pitch, loudness and timbre, we collected a simple test data set performed by our domain experts: we recorded 4 music scales, ranging from solfège Do in ocave 4 to So in ocave 5 in D major, played by Bamboo Flute, Erhu, Pipa and Zheng. We colored the points of Bamboo flute as blue, Erhu as orange, Pipa as green and Zheng as red. We have 12 pitches for each instrument. Pitch clusters are labeled by texts, such as Do4, Re4 and Do5, etc. For traditional Chinese music, the convention is not to label the pitch by letters, e.g. D4, E4 and D5, etc., because musicians who play traditional Chinese music read numbered musical notations and sing the solfège. The recording of each musical note lasted for 0.25 seconds, so each musical note has 10 note samples after being converted to the spectrogram. We also reproduced the recording to 4 loudness levels (marked as l1, l2, l3 and l4 in the hanging over label).

To get the segment latent vectors of traditional Chinese music spectrograms, we first subdivided each spectrogram successively into ten-second segments across the time axis and discarded the remaining part that is less than ten seconds. For each ten-second segment, we took every 20 columns of spectrogram along the time axis resulting in a downsampled spectrogram data of 20 columns with 501 dimensions for each column. Then the downsampled spectrogram was fed into our pretrained fully-connected Autoencoder's encoder, column by col-

umn, to extract the corresponding note latent vectors. These 20 note latent vectors were taken as the input for LSTM Autoencoder's encoder to get a segment latent vector of 16 dimensions.

After that, we had the following evaluation tasks to test the effectiveness of the extracted latent vectors:

T1. Verify if the note latent vectors can capture pitch, loudness and instruments information. Beautiful melody is always the first thing to be remembered when people listen to music. Pitch precision guarantees the beauty of melody. We clustered both note samples and note latent vectors to test if pitch variations are encoded into the note latent vectors. Our comparison panel (Figure 7) shows the difference and similarity between the note samples and the note latent vectors. On the panel of note samples (the left panel in Figure 7), we can observe that the same note played by the same instruments (e.g. cluster Fa4 on the left panel shown in Figure 7 (A)) are the most prominent clusters, but the same note played by different instruments are not clustered to an obvious group (i.e. not all identical colors form obvious clusters). On the panel of the note latent vectors (the right panel in Figure 7), we still notice the clusters of the same note played by the same instruments and they are tighter (e.g. cluster Fa4 shown in Figure 7(B)); moreover, we can even recognize some more distinct groups of the same pitch notes played by different instruments (e.g. the obvious group of So shown in Figure 7(C)). For example, there are large pitch groups of Do, Re, Mi, So, and La containing all four instruments. Hence, the note latent vectors are more sensitive to pitch compared to the original note samples. Still, we notice the Fa and Ti do not form a tight pitch group as the other pitches; however, the domain expert said that Fa and Ti were always played differently among performers because traditional Chinese music, known as pentatonic scale music, did not give a definitive pitch of Fa and Ti. Thus, we can reach a conclusion that the note latent vectors are able to acquire the pitch variations and make the pitch feature distinct.

The second property of note that we observed is the loudness of notes, variation of which changes the emotions in music. On both the panels we can recognize some small clusters of four loudness levels. For example, Fa4 cluster of Erhu on both sides shown in Figure 7(A) and Figure 7(B) has four distinct loudness clusters; however, we cannot conclude that if the note latent vectors enhance the difference of the loudness or not, since we can find that some notes such as Do5 of Erhu shown in Figure 7(D) and Re4 of Bamboo Flute shown in Figure 7(E) have more clear subclusters of loudness, some notes such as Fa4 and Mi5 of Pipa shown in Figure 7(F) and Figure 7(G) do not preserve the clear subclusters in the note latent projection. Nevertheless, it is still clear that the note latent vectors encode some loudness information.

The third property of note is timbre, from which people with some music background can differentiate instruments. Both panels do not have obvious big clusters of certain instruments, but we notice that none of the note samples or note latent vectors with the same pitch are overlapped, i.e. no points with different colors are overlapped. Thus, the note latent vectors also encode the timbre information, although it is the pitch not timbre dominate the separation of clusters.

T2. Verify if the segment latent vectors can discriminate different musical expressions. One of the most essential aspect of musical performance is how the performer expresses music. Since music is an expression of performers' emotional states, even if they are playing the same song with the same instrument, there are still differences in their musical performance such as adoption of accompaniment.

In MusicLatentVIS, we selected "Guang Ming Xing" from the drop-down menu of the music names which resulted in a subset contains two performers (Shuai Chai and Jun Rong) performing the same music with Erhu. As we mentioned before, this t-SNE projection is a 2D visualization of n-dimensional ($n = 16$) segment latent vectors. Each point in t-SNE represents a music segment of ten seconds. We applied Hierarchical Clustering with the euclidean distance as the measure of similarity, two as the number of clusters and ward linkage (a.k.a. minimum variance method which tries to minimize the intra-cluster variance when merging two clusters) on the selected latent vectors. After that, we can see the corresponding points are highlighted and colored by clustering result with one cluster in orange and the other in blue as shown in Figure 8(A). Orange cluster is denser. Out of curiosity, we re-colored these data samples by performers in Figure 8(B) where the purple cluster is Shuai Chai and the yellow cluster is Jun Rong. Surprisingly, we notice they are consistent with our previous Hierarchical Clustering result. Their heatmap and parallel coordinate are generated in Figure 8(C) and Figure 8(D). By brushing on the parallel coordinates of the latent dimensions which has the same color scheme as Figure 8(B), we found that both dim-1 (i.e., the second highest variance dimension) and dim-10 can completely separate these two clusters as shown in Figure 8(E) and Figure 8(F).

With the assistance of our domain experts, we notice that although the music played by the two performers has the same rhythm, they are different from the accompaniment: Chai's has accompaniment but Rong's does not, since for this song, Rong expected to prompt the beauty of Erhu particularly. That might also be the reason why Chai's cluster is sparser, which means Chai's has more variations among different segments of music. Since adoption of accompaniment is a part of performer's expression of music, we conclude that the segment latent vectors can discriminate different expressions of music to some extent.

T3. Verify if the segment latent vectors can indicate changes of musical pattern. Music's pattern here refers to a duration of repeated rhythm or dynamic (variations in loudness) which is another critical element to describe and retrieve music information. To achieve our verification goal, we selected two songs named "Tai Wan Feng Qing" and "Fen Hong Lian" from our dataset. The former one is played by performer Jinlong Fang with Pipa while the latter one is played by performer Ji Qiu with Zheng. The length of "Tai Wan Feng Qing" is 328 seconds, so there are totally 32 points in t-SNE projection for "Tai Wan Feng Qing". Similarly, "Fen Hong Lian" has 34 points.

We sorted the 10-second segment latent vectors by time and generated heatmaps of their cosine similarity as shown in Figure 9. As we discussed before, the cosine similarity is calculated based on the segment latent vectors extracted from our

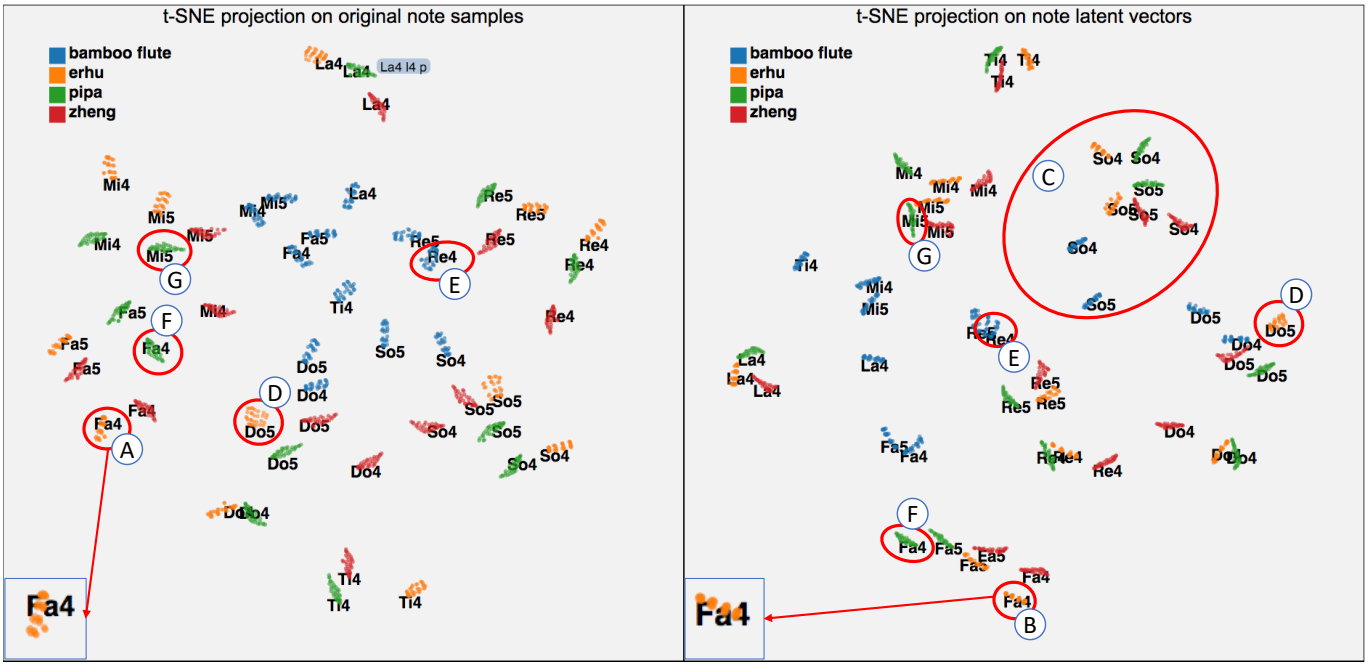

Fig. 7. The left panel is for the t-SNE projection of note samples and the right panel is for the t-SNE projection of note latent vectors.

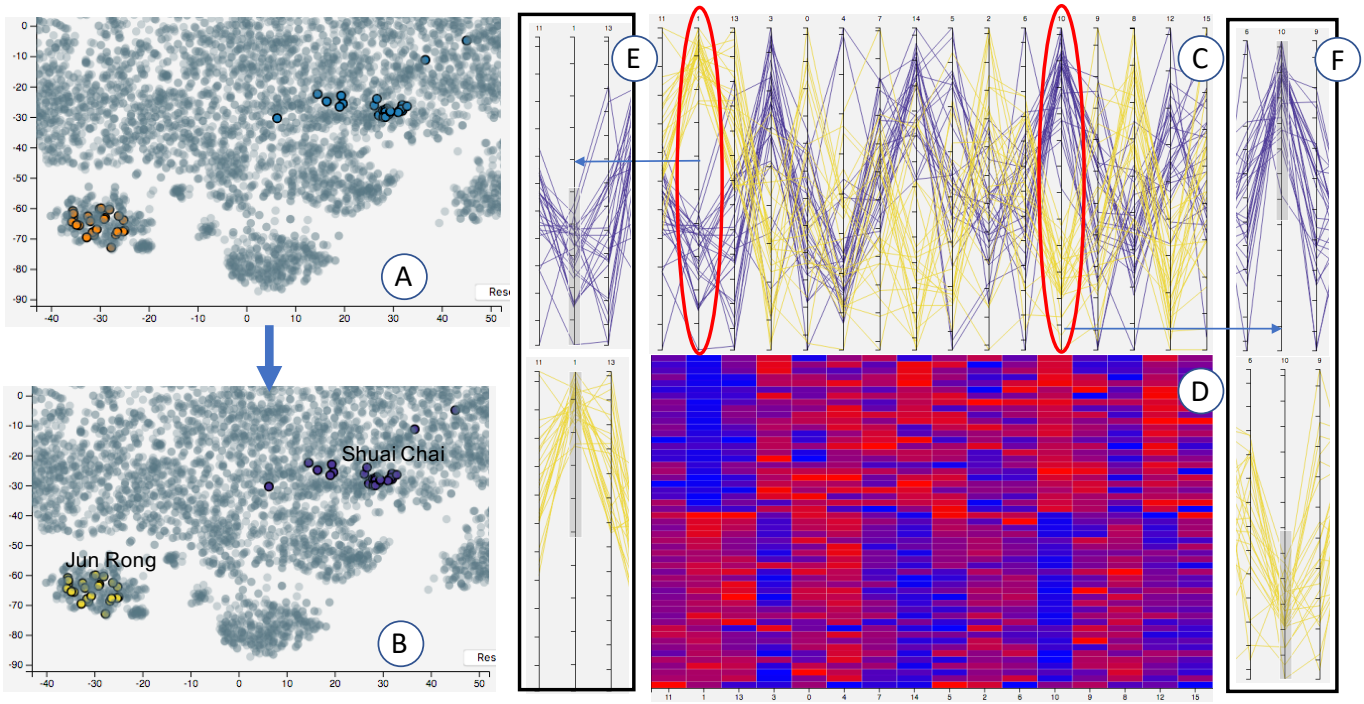

Fig. 8. t-SNE projection and clustering result of music: Guang Ming Xing. Each highlighted point in t-SNE is a 10-second segment of this music. (A) is the clustering result. Selected data point are colored by clusters. (B) shows the clusters colored by actual performers. By brushing on the parallel coordinates plot in (C), two dimensions (dim-1 and dim-10) are identified which can completely separate performers. (D) is the heatmap of latent vectors sorted by dimension variance. Each row in (D) represents a latent vector of 16 dimensions.

LSTM Autoencoder. If we select N music segments, after considering each pair of these segments, we will have a cosine similarity matrix of size $N \times N$. The (i, j)-th value in this matrix indicates the cosine similarity between the i-th and the j-th vector. Since each vector represents a duration of ten seconds, actually the (i, j)-th value is the similarity between the two music segments: one is the i-th segment from $i\times10$-th to $(i+1)\times10$-th seconds, another is the j-th segment from $j\times10$-th to $(j+1)\times10$-th seconds. The color at position (i, j) is the similarity between the i-th segment and the j-th segment. The more similar, the brighter it will be. We can visually assess which pair of segments are similar / dissimilar and hence can recognize the temporal patterns. A pattern on the diagonal of the similarity matrix means a continuous musical pattern while a pattern on the off-diagonal means two repeated patterns (repeat in time).

"Tai Wan Feng Qing" has 32 points in total, so its cosine similarity matrix is a 32×32 matrix as shown on the left side of Figure 9. We noticed a brighter square from the 12th segment to the 14th segment as shown in Figure 9(A). This indicates a consistency within a duration of 30 seconds (i.e., from 120 to 150 seconds). In addition, we found a pattern from the 18th segment to the 22th segment in Figure 9(B), our domain experts verified that a sudden speed change happens at around the 180th second and this period (from 180 to 230 seconds) has a faster rhythm than before and after. The pattern from the 25th to the 26th segment in Figure 9(C) is a 20-second duration of soft and slow rhythm. We can also locate a repeat pattern in Figure 9(D) which means the 29th segment has very similar behavior comparing to the 30th segment. This can be seen from the same color variation pattern along the horizontal stripe highlighted in Figure 9(D) between the two segments. When listening to the music, we realized there was a uniform soft style during the time period.

As for "Fen Hong Lian" on the right side of Figure 9, we can clearly see many highly related music segments (e.g., segments from 50 to 120 seconds and segments from 120 to 160 seconds as shown in Figure 9(G) and Figure 9(H)). The domain experts located several segments (e.g., the segments from 10 to 30 seconds and segments from 120 to 150 seconds in Figure 9(E) and Figure 9(H)) are "slient" segments, which means they only consist of several separate notes instead of forming a meaningful sentence. Since the segments from 30 to 50 seconds (as shown in Figure 9(F)) have strongly related musical notes, when comparing with the segments from 10 to 30 seconds and 120 to 150 seconds, they have low similarity values (darker color) in Figure 9(I) and Figure 9(K). In contrast, these "silent" segments have relatively high similarity values (brighter color) which can be seen in Figure 9(J).

From the observations above, we conclude that patterns identified from the segment latent vectors agree with the actual musical patterns.

5. Model Training

To get spectrograms with enough information of music, we sampled each music at the rate of 20kHz, which means the sampled wave has at least two samples for any cycles of frequency

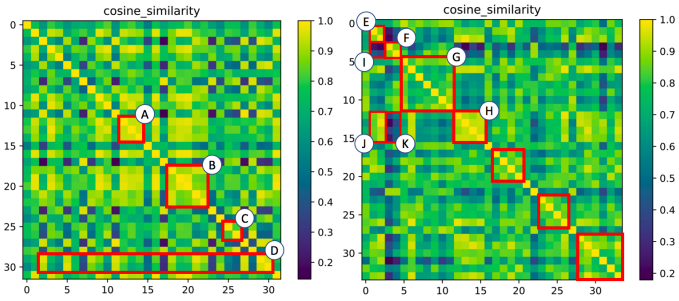

Fig. 9. Cosine similarity heatmaps of music. The left side is "Tai Wan Feng Qing" and the right side is "Fen Hong Lian". Each row i is the cosine similarity between the i-th segment and other segments. The cosine similarity values are encoded by colors (the brighter the more similar).

below 10kHz. The frequency range for traditional Chinese instruments is from 41.2 Hz to 2.8 kHz, so 0 ~ 10kHz is enough for a musical note (both fundamental frequency and overtones). When applying the short-time Fourier Transform, we adopted a Hanning window of length 500 so the window can cover a sound wave of frequency as low as 40Hz.

When training the fully-connected Autoencoder, we randomly selected 160 columns from each spectrogram of all 324 songs as we mentioned in Section 4. With a batch size of 512, learning rate of 0.001, and Adam stochastic optimizer, we reached a satisfied loss at epoch 500 as shown in left part of Figure 10. The training time for fully-connected Autoencoder is about 17 minutes on 129,600 501-dimensional note samples for 500 epochs.

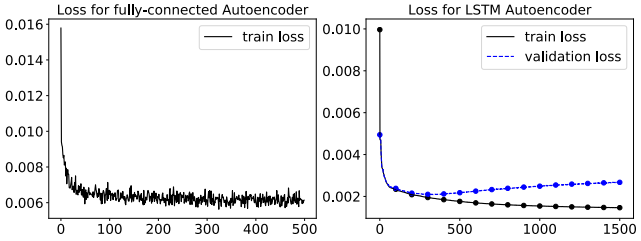

Fig. 10. The left side is the training loss for fully-connected Autoencoder with batch size of 512 and learning rate of 0.001. The right side is the training loss and validation loss for LSTM Autoencoder with batch size of 128, learning rate of 0.001 and dropout rate of 0.5 to avoid overfitting.

For LSTM Autoencoder, we collected a dataset of size 6480 for training and another dataset of size 1620 for validation. The training data were constructed by randomly chunking ten-second slices from each music spectrogram for 20 times (chunk 5 times for validation) and then down-sampled and fed them into the well-trained fully-connected Autoencoder. The collection of the latent vectors from the fully-connected Autoencoder are the training samples we need for the LSTM model.

With a batch size of 128, learning rate of 0.001, dropout rate of 0.5 to avoid over-fitting and using Adam for stochastic optimization, after training for 300 epochs, we obtained a validation loss of 0.00210. The training loss and validation loss are shown in right part of Figure 10. The training time for LSTM Autoencoder is about 3.5 hours on 129,600 note latent vectors for 1500 epochs.

6. Conclusion

With the help of deep neural networks for music information retrieval (MIR) and performance comparison on traditional Chinese music, we proposed a visual analysis system to evaluate the effectiveness of learned features. To acquire proper training data for our models, we converted from a traditional Chinese music CD collection to a content-rich spectrogram dataset. We utilized two variations of Autoencoders to extract features from their latent space: a fully-connected Autoencoder for note latent vector extraction (features without temporal correlation) and a LSTM Autoencoder for segment latent vectors (features with temporal correlation). Our interactive system allows users to investigate the distribution, clustering and similarity comparison of selected latent vectors. We performed several evaluation tasks through our system to demonstrate the usefulness of our system.

In future work, we would like to extent the scope of our studies to include a wider range of instruments as well as contemporary Chinese music that has an increased focus on harmonies. We will also explore supervised deep learning models for feature extraction since a supervised task can generate more relevant and controlled latent representations. Besides, our final goal is to facilitate music performance comparison and information retrieval. After proper evaluation, once we can confirm that the latent representations are effective, we can use them to conduct further visual analysis and build a comprehensive Chinese music information retrieval system. Also, as a complementary for the evaluation of these latent representations, we can use these representations to do classification or clustering and compare the performance with the original data. For further studies on other types of audio data with semantic meanings (e.g., western music, speech, or even natural sounds), a more general system is needed.

References

Agostini, G., Longari, M., Pollastri, E., 2003. Musical instrument timbres classification with spectral features. EURASIP Journal on Advances in Signal Processing 2003, 943279. URL: https://doi.org/10.1155/S1110865703210118, doi:10.1155/S1110865703210118.

Camargo, J.E., González, F.A., 2014a. Multimodal visualization based on latent topic analysis, in: 2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), pp. 1–6. doi:10.1109/ICMEW.2014.6890716.

Camargo, J.E., González, F.A., 2014b. Multimodal visualization based on latent topic analysis, in: 2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), pp. 1–6. doi:10.1109/ICMEW.2014.6890716.

Changsheng Xu, Maddage, N.C., Xi Shao, Fang Cao, Qi Tian, 2003. Musical genre classification using support vector machines, in: 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)., pp. V–429. doi:10.1109/ICASSP.2003.1199998.

Choi, K., Fazekas, G., Sandler, M.B., 2016. Automatic tagging using deep convolutional neural networks. CoRR abs/1606.00298. URL: http://arxiv.org/abs/1606.00298, arXiv:1606.00298.

Chuan, C.H., Agres, K., Herremans, D., 2018. From context to concept: exploring semantic relationships in music with word2vec. Neural Computing and Applications URL: https://doi.org/10.1007/s00521-018-3923-1, doi:10.1007/s00521-018-3923-1.

Cooper, M., Foote, J., Pampalk, E., Tzanetakis, G., 2006. Visualization in audio-based music information retrieval. Computer Music Journal 30, 42–62. doi:10.1162/comj.2006.30.2.42.

Costa, Y.M., Oliveira, L.S., Silla, C.N., 2017. An evaluation of convolutional neural networks for music classification using spectrograms. Applied Soft Computing 52, 28 – 38. URL: http://www.sciencedirect.com/science/ARTICLE/pii/S1568494616306421, doi:https://doi.org/10.1016/j.asoc.2016.12.024.

Cruz, L., Rolla, V., Kestenberg, J., Velho, L., 2018. Visual representations for music understanding improvement, in: Aramaki, M., Davies, M.E.P., Kronland-Martinet, R., Ystad, S. (Eds.), Music Technology with Swing, Springer International Publishing, Cham. pp. 468–476.

Dieleman, S., Schrauwen, B., 2013. Multiscale approaches to music audio feature learning, in: ISMIR.

Dieleman, S., Schrauwen, B., 2014. End-to-end learning for music audio, in: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6964–6968.

Downie, J.S., . Music information retrieval. Annual Review of Information Science and Technology 37, 295–340. URL: https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/aris.1440370108, doi:10.1002/aris.1440370108.

Downie, J.S., 2004. The scientific evaluation of music information retrieval systems: Foundations and future. Computer Music Journal 28, 12–23. URL: https://doi.org/10.1162/014892604323112211, doi:10.1162/014892604323112211, arXiv:https://doi.org/10.1162/014892604323112211.

Eck, D., Schmidhuber, J., 2002. Finding temporal structure in music: blues improvisation with lstm recurrent networks, in: Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing, pp. 747–756. doi:10.1109/NNSP.2002.1030094.

Foote, J., 1999. Visualizing music and audio using self-similarity, in: Proceedings of the Seventh ACM International Conference on Multimedia (Part 1), ACM, New York, NY, USA. pp. 77–80. doi:10.1145/319463.319472.

Garcia-Gasulla, D., Béjar, J., Cortés, U., Ayguadé, E., Labarta, J., 2015. Extracting visual patterns from deep learning representations. CoRR abs/1507.08818. URL: http://arxiv.org/abs/1507.08818, arXiv:1507.08818.

Hristov, Y., Lascarides, A., Ramamoorthy, S., 2018. Interpretable latent spaces for learning from demonstration. CoRR abs/1807.06583. URL: http://arxiv.org/abs/1807.06583, arXiv:1807.06583.

Janssen, B., van Kranenburg, P., Volk, A., 2017. Finding occurrences of melodic segments in folk songs employing symbolic similarity measures. Journal of New Music Research 46, 118–134. URL: https://doi.org/10.1080/09298215.2017.1316292, doi:10.1080/09298215.2017.1316292, arXiv:https://doi.org/10.1080/09298215.2017.1316292.

Ji, X., Shen, H., Ritter, A., Machiraju, R., Yen, P., 2019. Visual exploration of neural document embedding in information retrieval: Semantics and feature selection. IEEE Transactions on Visualization and Computer Graphics 25, 2181–2192. doi:10.1109/TVCG.2019.2903946.

Jiang, D.N., Lu, L., Zhang, H., Tao, J., Cai, L., 2002. Music type classification by spectral contrast feature. Proceedings. IEEE International Conference on Multimedia and Expo 1, 113–116 vol.1.

Kim, H., Kim, B., Zhang, B., 2009. Evolutionary hypernetworks for learning to generate music from examples, in: 2009 IEEE International Conference on Fuzzy Systems, pp. 47–52. doi:10.1109/FUZZY.2009.5277047.

Koenigstein, N., Dror, G., Koren, Y., 2011. Yahoo! music recommendations: Modeling music ratings with temporal dynamics and item taxonomy, in: Proceedings of the Fifth ACM Conference on Recommender Systems, ACM, New York, NY, USA. pp. 165–172. URL: http://doi.acm.org/10.1145/2043932.2043964, doi:10.1145/2043932.2043964.

Korzeniowski, F., Widmer, G., 2016. A fully convolutional deep auditory model for musical chord recognition. CoRR abs/1612.05082. URL: http://arxiv.org/abs/1612.05082, arXiv:1612.05082.

Kumarbanchhor, S., Khan, A., 2012. Musical instrument recognition using zero crossing rate and short-time energy. International Journal of Applied Information Systems 1, 16–19. doi:10.5120/ijais12-450131.

Laden, B., Keefe, D.H., 1989. The representation of pitch in a neural net model of chord classification. Computer Music Journal 13, 12–26. URL: http://www.jstor.org/stable/3679550.

Lee, J., Park, J., Kim, K.L., Nam, J., 2017. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. CoRR abs/1703.01789. URL: http://arxiv.org/abs/1703.01789, arXiv:1703.01789.

Li, X., Xianyu, H., Tian, J., Chen, W., Meng, F., Xu, M., Cai, L., 2016. A deep

bidirectional long short-term memory based multi-scale approach for music dynamic emotion prediction, in: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 544–548. doi:10.1109/ICASSP.2016.7471734.

Liu, S., Bremer, P., Thiagarajan, J.J., Srikumar, V., Wang, B., Livnat, Y., Pascucci, V., 2018. Visual exploration of semantic relationships in neural word embeddings. IEEE Transactions on Visualization and Computer Graphics 24, 553–562. doi:10.1109/TVCG.2017.2745141.

Liu, Y., Jun, E., Li, Q., Heer, J., 2019. Latent space cartography: Visual analysis of vector space embeddings. Comput. Graph. Forum 38, 67–78.

Marchi, E., Vesperini, F., Eyben, F., Squartini, S., Schuller, B., 2015. A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional lstm neural networks, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1996–2000. doi:10.1109/ICASSP.2015.7178320.

McFee, B., Barrington, L., Lanckriet, G.R.G., 2011. Learning content similarity for music recommendation. CoRR abs/1105.2344. URL: http://arxiv.org/abs/1105.2344, arXiv:1105.2344.

Meyer, M., Beutel, J., Thiele, L., 2017. Unsupervised feature learning for audio analysis. CoRR abs/1712.03835. URL: http://arxiv.org/abs/1712.03835, arXiv:1712.03835.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality, in: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems 26. Curran Associates, Inc., pp. 3111–3119.

Muller, M., Ellis, D.P.W., Klapuri, A., Richard, G., 2011. Signal processing for music analysis. IEEE Journal of Selected Topics in Signal Processing 5, 1088–1110. doi:10.1109/JSTSP.2011.2112333.

Nabney, I.T., Sun, Y., Tino, P., Kaban, A., 2005. Semisupervised learning of hierarchical latent trait models for data visualization. IEEE Transactions on Knowledge and Data Engineering 17, 384–400. doi:10.1109/TKDE.2005.49.

Nam, J., Herrera, J., Slaney, M., Smith, J.O., 2012. Learning sparse feature representations for music annotation and retrieval, in: ISMIR.

van den Oord, A., Dieleman, S., Schrauwen, B., 2013. Deep content-based music recommendation, in: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems 26. Curran Associates, Inc., pp. 2643–2651. URL: http://papers.nips.cc/paper/5004-deep-content-based-music-recommendation.pdf.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A.W., Kavukcuoglu, K., 2016. Wavenet: A generative model for raw audio. CoRR abs/1609.03499. URL: http://arxiv.org/abs/1609.03499, arXiv:1609.03499.

Panda, S., Namboodiri, V.P., Roy, S.T., 2019. Visualizing music genres using a topic model. URL: https://doi.org/10.5281/zenodo.3249352, doi:10.5281/zenodo.3249352.

Peeters, G., 2004. A large set of audio features for sound description (similarity and classification) in the cuidado project .

Schluter, J., Osendorfer, C., 2011. Music similarity estimation with the mean-covariance restricted boltzmann machine, in: 2011 10th International Conference on Machine Learning and Applications and Workshops, pp. 118–123. doi:10.1109/ICMLA.2011.102.

Sigtia, S., Benetos, E., Boulanger-Lewandowski, N., Weyde, T., d'Avila Garcez, A.S., Dixon, S., 2015. A hybrid recurrent neural network for music transcription, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2061–2065. doi:10.1109/ICASSP.2015.7178333.

Slaney, M., Weinberger, K., White, W., 2008. Learning a metric for music similarity., pp. 313–318.

Srivastava, N., Mansimov, E., Salakhutdinov, R., 2015. Unsupervised learning of video representations using lstms. CoRR abs/1502.04681. URL: http://arxiv.org/abs/1502.04681, arXiv:1502.04681.

Sutskever, I., Vinyals, O., Le, Q.V., 2014. Sequence to sequence learning with neural networks. CoRR abs/1409.3215. URL: http://arxiv.org/abs/1409.3215, arXiv:1409.3215.

Tang, T., Jia, J., Mao, H., 2018. Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis, in: Proceedings of the 26th ACM International Conference on Multimedia, ACM, New York, NY, USA. pp. 1598–1606. doi:10.1145/3240508.3240526.

West, K., Lamere, P., 2006. A model-based approach to constructing music similarity functions. EURASIP Journal on Advances in Signal Processing 2007, 024602. URL: https://doi.org/10.1155/2007/24602, doi:10.1155/2007/24602.

Wood, A., Bowsher, J., 1980. The Physics of Music. Greenwood Press. URL: https://books.google.com/books?id=nLQQAQAAMAAJ.

Yim, J.D., Shaw, C., Bartram, L., 2009. Musician map: Visualizing music collaborations over time, p. 72430. doi:10.1117/12.812377.

Yu, Y., Luo, S., Liu, S., Qiao, H., Liu, Y., Feng, L., 2020. Deep attention based music genre classification. Neurocomputing 372, 84 – 91. URL: http://www.sciencedirect.com/science/ARTICLE/pii/S0925231219313220, doi:https://doi.org/10.1016/j.neucom.2019.09.054.

Zeiler, M.D., Fergus, R., 2013. Visualizing and understanding convolutional networks. CoRR abs/1311.2901. URL: http://arxiv.org/abs/1311.2901, arXiv:1311.2901.

Zhao, F., Feng, J., Zhao, J., Yang, W., Yan, S., 2018. Robust lstm-autoencoders for face de-occlusion in the wild. IEEE Transactions on Image Processing 27, 778–790. doi:10.1109/TIP.2017.2771408.

Zhu, W., Chen, C., 2007. Storylines: Visual exploration and analysis in latent semantic spaces. Computers & Graphics 31, 338 – 349. URL: http://www.sciencedirect.com/science/ARTICLE/pii/S0097849307000568, doi:https://doi.org/10.1016/j.cag.2007.01.025.