CREATING MUSICAL PALETTE OF MELODIES: MELODIC SIMILARITY BASED ON INHERENT SEQUENTIAL FEATURE

Jung Choi  
MAClab, GSCT, KAIST 
kaka00@kaist.ac.kr

Chaelin Park  
MAClab, GSCT, KAIST 
lynn08@kaist.ac.kr

Halla Kim  
theQlab, GSCT, KAIST 
kimhalla@kaist.ac.kr

ABSTRACT

The strength of Autoencoder arises from its potential use for creative applications of machine learning. Measuring similarity between melodies is also what makes Autoencoder popular modeling method. We can evaluate the similarity based on the distance between latent feature vectors. With Nottingham Folk Song dataset, we converted melodies to melodic contour vector and experimented several settings of the models. Our goal is to construct a reliable framework that gives similarity measure that matches our perception. We could confirm our encoding method is valid and glimpse through the melodic movement pattern in music with our model.

1. INTRODUCTION

Creating artwork is a process of configuring intangible images that a creator has in his mind. For songwriters, it is thus natural that they often struggle to come up with appropriate melody which matches well with what they are trying to express. Our model enables creators to explore wide scope of new melodies only by inspecting latent space learned from the sample inputs, which leads to the concept of assisted creativity.

Measuring similarity can be the first step for this kind of assisted creativity in that it provides computational basis for musical intelligence. To assist human in terms of creativity in music, machine should be able to distinguish how much one melody is different from another to make musical decisions as humans do. Thus, it is important to design similarity measurement to match our perception. Music similarity measure is one of the significant areas of Music Information Retrieval (MIR) researches. Numerous approaches have been introduced with mostly knowledge-based engineering.

To find the model with highest-performance, we experimented a diverse settings of Autoencoder changing hyperparameters and architecture tunings. Although there exists biases of the dataset used, we confirmed that recurrent bi-directional learning is more appropriate for music sequence than the others we tested. We could figure out that the melodic movement information is well-embedded in compressed dimensions, thereby enabling to find similar melodies with given query based on this information.

2. BACKGROUNDS

2.1 Autoencoder

Autoencoder is one of well-known deep neural network generative models. This framework provides compressed expression of input data. Its strength is to explicitly model $p(z|x), p(z)$, and $p(x|z)$, where $z$ is a latent vector learned from the data.

![Figure 1. Poor reconstruction quality of VAE](image)

2.2 Recurrent bi-directional Autoencoder

Recurrent bi-directional Autoencoder is a modeling framework whose encoder and decoder is recurrent neural network, which is a popular machine learning framework for AE.

For the encoder and the decoder, two-layer bidirectional LSTM network (Hochreiter Schmidhuber, 1997; Schuster Palizal, 1997) is frequently used. Bidirectional structure is proper to adopt in learning musical sequence as the music usually sounds natural when both the former and the latter part of the sequence are connected smoothly. For this reason, it is better to consider the hidden features from either way of directions. Long Short-Term Memory (LSTM) is in effect the standard of recurrent neural network (RNN). Compared to RNN, it resolves the vanishing or exploding gradients problem posed by repetitive multiplication during back-propagation through time (BPTT). LSTM holds information in memory cells controlled by gates learned from the data.

The encoder of RNN framework makes hidden state
vectors h1, h2, ..., hT from an input sequence. The input sequence is given to bidirectional LSTM to obtain the two final state vectors. These vectors are then concatenated.

2.3 Variational Autoencoder

Variational Autoencoder (VAE) is a framework that provides compressed expression of input data using parameterized distribution space. Following the basic structure of Autoencoder, VAE consists of an encoder qθ(z|x) which approximates the posterior p(z|x) and a decoder pθ(x|z). In terms of Variational Inference, we do posterior inference by maximizing evidence lower bound (ELBO) which corresponds to minimizing KL divergence between an encoder, the approximate posterior and the prior, forces the approximate posterior to be close to the actual prior p(z). There are ways to control this trade-off, two of which are either to multiply KL divergence term or to set some threshold to KL divergence only to penalize this term when the divergence exceeds certain threshold. The latter method gives the model free bits.

2.4 Recurrent VAE

Recurrent VAE is a form of VAE with recurrent neural network in both encoder and decoder. The parameters of latent space distribution, μ and σ is parameterized by final state hidden vector. The decoder then learn pθ(x|z) by reconstructing the same input sequence using the latent code z.

\[
\begin{align*}
\mu &= W_{h\mu} h_T + b_{\mu} \\
\sigma &= \log(\exp(W_{h\sigma} h_T + b_{\sigma}) + 1 \\
\end{align*}
\]

where \( W_{h\mu}, W_{h\sigma} \) and \( b_{\mu}, b_{\sigma} \) are weight matrices and bias vectors, respectively.

When it comes to recurrent neural network, however, one of the problems when dealing with long input sequence is that the decoder aggressively outputs the sequence without considering latent code. In this case, the KL divergence is set to zero even though the encoder does not work. This is called `posterior collapse` where RNN autoregressively models the data, making hard for long sequences to be encoded as latent space vectors.

3. RELATED WORKS

There has been many experiments on morphing high-dimensional musical information into low-dimensional space representations.

Daniel V. Oppenheim[1] is known for building an interface that allows users to mix different up to four MIDI compositions by mixing different elements of each piece (such as rhythm, pitch, dynamics, timbre, harmony). Users could choose which elements to mix with checkboxes and also select interpolating methods.

Bretan et al. [2] could be mentioned as one of the standard experiments of melodic interpolation. With monophonic musical phrases, they construct a latent space using Autoencoder. By interpolating the nearest points in latent space, it gives gradual progression in both the harmonic and rhythmic elements.

Works on morphing different timbre includes the Tristan Jehan [3] which synthesize timbres of various musical instruments such as a flute or choral vocals. The performer could control the timbre while making physical movements along a path.

MusicVAE [4], one of the implementations of Google Magenta Project, proposed hierarchical decoder which shows better performance on long-term sequence. This model avoids posterior collapse which remains as main issue of recurrent VAE. It shows better sampling, reconstruction quality, and interpolation for long-term sequence.

4. EXPERIMENTS

4.1 Data and Training

We use Nottingham Folk Song for our dataset. This is a set of melodies from 1035 British folk dance tunes, available in ASCII, gif or pdf format. As shown in Figure 2, we converted the melodies of into our data scheme, melody contour vector. We excluded melodies with triplet beats and set 2-bar melody as a base unit of the data. After segmenting the tunes into 2-bar melodic sequence, we extracted a 2-dimensional vector (note duration, pitch interval) each. For example, if 8th note of C is followed by 4th note of G, the input vector becomes \((0.5, 7)\). This way, we obtain 15313 melodic sequences. The length of a vector varies depending on the number of notes the 2-bar sequence contains. Thus, in order to make sure all sequences have
the same length, we set the length as 32 encodings of (note duration, pitch interval) and applied zero-padding for short sequences. The final input is vector of (2 X 32) shape.

4.1.1 Model

We tested three kinds of models. Figure 3 describes the basic structure of Autoencoder, Variational Autoencoder, and Recurrent Autoencoder from the left.

Figure 3. Description of three models used: Autoencoder, Variational Autoencoder, and Recurrent Autoencoder

4.2 Results

We first processed our melody dataset with Variational Autoencoder consisting of fully-connected layers. The number of dimensions in latent space is 128. After training, however, we could not take the resulting latent space useful. As in Figure 4, the reconstruction quality of VAE was poor.

Partly because of the simplicity of the melodies in the original dataset, input sequences tend to converge to some point near the mean value of the Gaussian latent space. As the latent feature vectors are densely concentrated on the core, it is hard to reconstruct various ranges of inputs from closely-positioned points. This limitation derives from our dataset itself, thus we should not conclude VAE poorly model the musical sequence data. At least, we could find out that simplicity of the data can hinder the model from constructing ideal distribution of latent space.

Figure 4. Reconstruction of vector in VAE

Considering the simplicity of the data, we then applied Autoencoder with recurrent neural network. Compared to VAE which parameterize the latent code into Gaussian distribution, Recurrent Autoencoder gives more reasonable results. In Figure 5, it shows good reconstruction quality of latent vectors. In Figure 6, it generates the input melody almost the same. We conducted further experiments by applying Recurrent Autoencoder in order to reflect the sequential characteristics of the melody. It is natural that the result from Recurrent Autoencoder showed better performance of embedding similar melodies into close locations in vector space.

Figure 5. Reconstruction of latent vectors in Recurrent Autoencoder

Figure 6. Reconstruction of melody in Recurrent Autoencoder

Figure 7. Outputs of query melody (1)

Figure 7 shows the model response of a given melody input. After reducing the 128-dimensional latent vector into 2-dimensional vector through TSNE, we calculated Euclidean distance and listed 8 melodies whose distance from the query melody was the smallest. Figure 8 shows another example of the model response of a given melody input. After taking the same steps to calculate the distance, we obtained a list of melodies whose distance from the query melody was small.

5. CONCLUSION

Despite the high performance of Variational Autoencoder demonstrated by previous research in general, fully-connected or recurrent Autoencoder fits better to our
Unlike we expected, VAE did not converge well to produce good representing embeddings. Even with numerous attempts of hyperparameters and architecture tunings, it only converged to produce a general average area of melodic contours. This is due to our specific dataset used here. We assume that the simplicity of the melody in our dataset makes it difficult for VAE to find ideal latent distribution.

Both fully-connected and Recurrent Autoencoder showed substantially reliable results. When we reduce the number of dimensions in latent space from 128 to 32, there were little difference between two frameworks giving low reconstruction loss. Our contribution lies in encoding the input melody as a sequence vector that carries the information of melodic movement, and evaluate the similarity between melodies within embedding space of deep auto-encoded architecture. For future work, we plan to apply our model to more complex dataset and rap rhythmic sequence in order to get deep understanding on melodic and rhythmic structure.

6. REFERENCES


http://hdl.handle.net/1721.1/61543

https://magenta.tensorflow.org/music-vae