CLASSICAL COMPOSERS CLUSTERING

Huan Zhang, Jingyuan Xing†

†Carnegie Mellon University

An Important Problem

Over the course of classical music history, composers have developed distinctive styles while influencing each other. Low-level quantifiable features like intervals, rhythms, tonal characteristics may be not prominent in human perception of music, but significantly contributes to the style of different composers. Thus we attempt to cluster compositions by different composers by applying clustering algorithms to these features extracted from patterns in music.
Using a MIDI dataset containing 4702 pieces of compositions spanning over 20 composers in from 13^{th} to 20^{th} century, we extracted different latent representation of compositions for clustering, including methods like feature extraction, autoencoders, and convolutional autoencoders.

Limitation of Existing Methods

Most of the previous works that related to composition clustering, takes a *similarity metric* approach that first define a distance between pair of music or composers, and then cluster based on the distance. Little machine learning algorithms are involved.

1. Georges [1] defined the metric of centralised cosine measure to cluster the composers.
2. In [2], an information theoretic approach was applied to hierarchical clustering of the music. They did not extract specific features from music but viewed MIDI data as a whole, and utilized the normalized compression distance (NCD) as a similarity metric to cluster composition.
3. However, there exists multiple works in labeled composition classification that uses machine learning approach. Jain et al. [3] experimented with algorithms like SVM with extracted features, softmax regression.

Clustering Methods

In the project we experimented with several traditional machine learning clustering algorithms.

1. **KMeans** The K means model is performed based on the Expectation Maximization algorithm. We first guess some cluster center, and then assign points to their nearest cluster center, and calculate the cluster mean. The initial condition can be randomized and we also experimented with the PCA based initial condition.
2. **Spectral clustering** first transfer the data points into a similarity graph. For the adjacentcy matrix representation A of the graph, Laplacian matrix is computed, and then find the first (smallest) k eigenvalues (Spectrals) and eigenvectors. Then, k-means clustering algorithm is applied for the n data points, using their representation as each row in the eigenvectors matrix of $n \times k$.
3. **Mean-shift clustering** is done by iteratively shifting the points towards the mode with high density of data, to eventually form clusters. It starts with each point forming its own cluster, and requires a distance function to find the neighbors. And then mean-shift of the point was computed by finding the mean of all the neighbors,

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x) x_i}{\sum_{x_i \in N(x)} K(x_i - x)}$$

where $K(x_i - x)$ is a kernel function.

Latent Representation of Compositions

Music is an relative abstract art form, thus a large task we are addressing is how to obtain a good latent representation of musical composition, so that they are distinctive when clustering. In the project, we tried three different methods in representing music:

1. **Music Theory features** (776 dim)
2. **Audoencoder latent representation** (16 dim)
3. **Convolutional Autoencoder latent representation** (32 dim)

1.We used an open source MIDI analytics software, **jSymbolic**, which is able to extract music theory related statistics. From the theoretic contents, we are able to represent each piece of the composition as feature vectors and numerically represent the piece. The information includes: **Pitch Statistics**, **Melodic Intervals**, **Vertical Intervals** , **Rhythm and Tempo**, **Instrumentation**, **Voice Leading**.

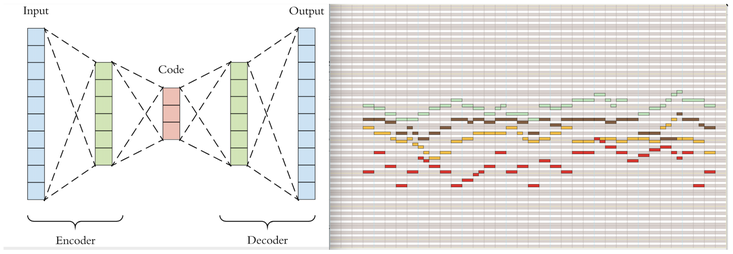


Fig. 1: Convolutional Autoencoder structure

2. We also tried to learn a latent representation of music using autoencoders. We first represent MIDI as a piano roll matrix - a 128 * 10000 matrix that pictures the note onset/offset, volume in the axis of time. 128 stands for the fact that there are 128 pitches in MIDI representation.
In this way, autoencoder plays a role of dimension reduction, that strives to replicate the original data from a compressed representation. In our case, a 16 dimension vector.
3. Building up on the previous autoencoder idea, we experimented with more complicated network structure. A convolution component come to mind as the piano roll matrix has a graphic-like interpretation. The structure is as pictured in the following diagram: 4 convolution layers (first 2 5x5 filter, second 2 3x3 filter) each with a maxpooling (2x2 filter) layer. Then the feature map was send into 3 dense layers that reduce to a representation 32-dim vector. As it's a convolutional autoencoder, the rest of layers mirrors the previous ones. The training objective, is again, reproducing the original piano roll representation.

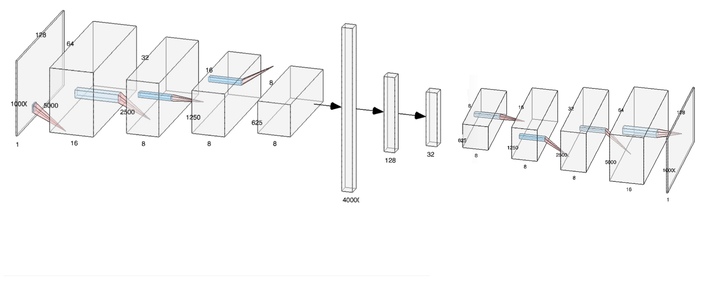


Fig. 2: autoencoder structure, and midi piano roll representation

Results

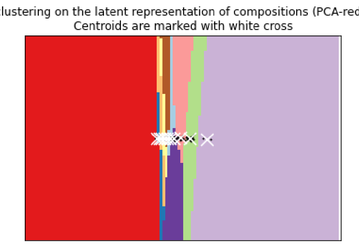
We use the standard unsupervised evaluation metric and protocols for evaluation. For all the clustering algorithms that we used, let the number of clusters be the number of ground-truth categories and evaluate performance with *unsupervised clustering accuracy (ACC)*, which takes a cluster assignment from an *unsupervised* algorithm and a ground truth assignment and then finds the best matching between them.ACC is defined as following:

$$ACC = max_m \frac{\sum_{i=1}^{n} 1\{l_i = m(c_i)\}}{n}$$

where l_i is the ground-trugh label, c_i is the clustering assignment produced by the algorithm, and m ranges over all possible one-to-one mappings between clusters and labels.

model	Pairwise	Contrasting Composers	Similar Composers
raw jSymbolic	0.78	0.58	0.52
ae jSymbolic	0.70	0.38	0.36
raw pianoroll	0.71	0.56	0.41
cae pianoroll	0.67	0.46	0.33

We also visualized the image of the clustering result (decision boundries of PCA reduced data) of 10 different composers, with a variety of compositions ranging from piano solo to symphonies. The v-measure of 10 composers clustering only achieved 0.172, and from the plot we can see that the representation (PCA reduced) of compositions are still quite homogenious in the middle of the plot.



Future Works

With the deep clustering and latent representation idea of music in mind, we can take a step further to induce more composer-specific representation. For example, we can first train a classification CNN on the data for different composers, and use the output as the second-to-last layer as the latent representation for clustering. As the training objective is to differentiate between composers, the representation should be more helpful in clustering.

References

[1] P. Georges and N. Nguyen. "Visualizing music similarity: clustering and mapping 500 classical music composers." In: *Scientometrics* 120 (2019), pp. 975–1003.

[2] P. Vitanyi R. Cilibrasi and R. de Wolf. "Algorithmic clustering of music." In: *Proceedings of the Fourth International Conference onWeb Delivering of Music* (2004), pp. 110–117.

[3] T. Yngesjo S. Jain A. Smit. "Analysis and Classification of Symbolic Western Classical Music by Composer." In: ().