# Embedding and Prediction of Musical Notes using Skip-gram Model

: with classical genre

Park Seongmin

## 1. Introduction

Natural Language Processing (NLP) is a sort of Artificial Intelligence (AI) that process and analyzes the text data using Machine Learning (ML). NLP uses ML algorithm to recognize structure of text data and its meaning. NLP can help users to understand more about text data and apprehend relation between context. It is used to analyze customer's emotion, classifying contents, medical field and etc.

Skip-gram model is one of NLP algorithm that predicting context words with regard to given words. It predicts context words with conditional probability function by what word is given as input. To analyze data with a skip-gram model, there are several conditions that data should satisfy.

First, data should be discrete rather than continuous since skip-gram model is deserve for analyzing categorical data such as text (corpus). It is not useful for continuous data because context of continuous data must have immediate predecessor and immediate successor as neighbor step, by definition of continuous function. Secondly, data should be relevant between contexts. Since skip-gram uses conditional probability function based on context, incoherent data will lead to meaningless result.

Inspired by such conditions, I came up with an idea that the melody of music can be regarded as text data and musical note data might be therefore adequate for skip-gram processing. We will deal with notes instead of words, and bars instead of sentences. Melody will used as corpus. Skip-gram algorithm will train with each piece of classical music by era (such as Baroque, Classicism, Romanticism), and predict context notes according to given input note. Moreover, based on the predictions provided by the

Skip-gram model, we will make a verse of the song that fits each age for trial. Finally, we do visualize each composer's song and embedding matrix by t-SNE.

# 2. Preparation for Analysis

## 2.1 Theory of Harmony

Musical notes are context-based data. However, they should be placed by a set of rules according to a melody, harmony and beat. Notes represent not only height of the pitch, but also length of the beat. The pitch is varies depending on which location(beat) the note appears. To be specific, in 4/4 beat for example, the first beat usually has the strongest accent and the third beat is then strong. These beats usually have pitches as tonic, for example in C Major scale, C(Do), E(Mi), G(sol) usually appears in first and third beat. Beats and pitches have a great deal of correlation. Those have to be analyzed interdependently, so note will be dataized with not only pitch but also beat. (That is very meaning of notes.)

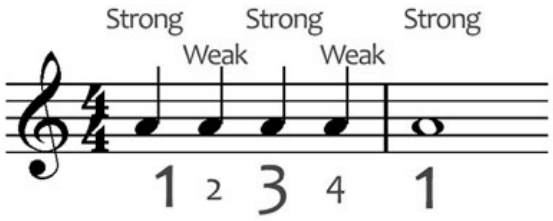



**Figure 1. Distribution of dynamics in beat 4/4**

However, all of songs does not have the same scale. Different scales have different notes in tonic. In D Major scale, D(Re), F#(Pa#), A(La) is of tonic notes. It will disrupt training on data, so we should standardize all of scales as C Major scale. It is impossible to change minor scale to major scale. Therefore, we only use major scale song to analyze.

A series of rules that apply to music appear differently from time to time. The Baroque period is when music techniques began to be established. Musicians of that era such as Johann Sebastian Bach (1685~1750) began to express emotions by establishing concepts such as dynamic, flow of melody and beat. Classicism is ancient Greek and Roman artistic trend in pursuit of harmony, uniformity, and brightness. Ludwig van Beethoven (1770~1827) is was a leading composer of that era. Romanticism made a move to break away from classicism. Musicians of those days sought freedom away from the classical form and sought to express emotions as they were. Robert Alexander Schumann (1810~1856) is one of the romanticism musicians.

Those three musical eras will be used in training. Skip-gram algorithm will give three different outcomes with regarding to era of music used in training.

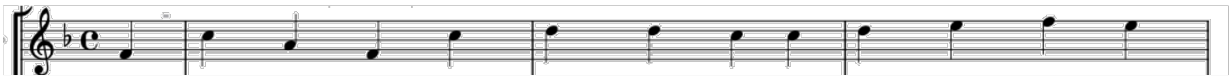
## 2.2 Data Explanation / Preprocessing

We will proceed analysis with python3 on Jupyter Notebook. Music21 is a Python-based toolkit for computer-aided musicology. People use music21 to answer questions from musicology using computers, to study large datasets of music, to generate musical examples, to teach fundamentals of music theory, to edit musical notation, study music and the brain, and to compose music.

It can deal with data extension of .MIDI, .mxl, and etc. An MXL file is a compressed music score created with MuseScore, a program for music composition and notation. It contains a compressed file saved in the MusicXML format. We will use mxl file to analyze data which is attached in library.

Among mxl files, we will select pieces of Bach, Beethoven and Schumann. After we import files, we have to filter out major scale songs and transpose them to C Major scale. Songs consists of several instruments/parts such as Soprano, Alto, Bass and Violin, Viola, Cello etc. We only scope with melody, so Soprano or Violin part will be selected only since they usually have a role of main melody. If we subtract melody, we need to make data to be suitable for analysis.

Each of note has information of pitch and beat. Information on which line is located among the five lines represents a pitch. The number of and shape of the notes represents length of beat. We will divide each bar into beats and indicate where the notes are located in beats. For example, figure below is piece of Bach and we labeled each note with ('pitch', 'location').



('C4', 4.0) ('G4', 1.0) ('E4', 2.0) ('C4', 3.0) ('G4', 4.0) ('A4', 1.0) ('A4', 2.0) ('G4', 3.0) ('G4', 4.0) ('A4', 1.0) ('B4', 2.0) ('C5', 3.0) ('B4', 4.0)

**Figure 2. Sequence of notes in Bach's piece and corresponding indication**

# 3. Problem Definition

The melodies of each piece are stored as a series of corpus. Each note corresponds to word with regard to text data, and such note data has a form as ('pitch', 'location'). We have to vectorize data, since computer cannot deal with category data as it is. So, we use one-hot encoding to be used in input data. And then we use skip-gram algorithm with window size 2. Such window size is chosen because too long sequence is meaningless for likelihood of notes and only with the sense that it would be appropriate.

We train the model by responding to each composer separately, for example, one model trained by Bach's pieces, another by Beethoven's and the other one by

Schumann's. For each model, we find the appropriate number of epochs which minimizes the loss. Then, with trained model, we predict adequate notes in order starting from ('C4','1.0'), tonic pitch C4 on the starting point. Each model corresponding to era will bear different prediction.

For visualization, we're going to create an adjacency matrix with corpus-like data. If two of notes lie in a line-to-follow relationship, the value of the adjacency matrix of predecessor index is added by 1. Depending on which era of trend was applied to the song, the pattern of adjacent matrices derived from the data will vary. Adjacency matrix is visualized by heatmap.

Finally, we use t-SNE model to observe embedded weight matrix (input to hidden). We will observe the t-SNE plot by giving a specific color to the index corresponding to certain beat, and we will do the same for the index corresponding to the tonic pitch.

# 4. Skip-gram Model Description

## 4.1 Dense Representation

One-hot encoding, also called as sparse representation, expresses the number of all cases a feature can have in each independent dimension. For example, if we have N kinds of words in corpus, then we make vector as N dimension. And only one of corresponding word has value 1, but 0 on otherwise. It is called sparse matrix because most of the components of the one-hot encoding vector are zero.

On the other hand, dense representation, also called as distributed representation, doesn't express each word as independent dimension. Instead, we express the object in correspondence with the number of dimensions we set. Information of word distributes various elements of vector. Several dimensions are combined to express the attributes to be expressed.

Dense presentation has the advantage of being able to express an object in a small dimension. Representing an object with sparse presentation usually increases the number of dimensions. It takes thousands of dimensions to express these words in sparse presentation. Furthermore, most of the vectors produced in this way have zero values. Then problem called a curse of dimensionality occurs. It becomes difficult to extract information from the data. And dense repression can express words that are related to each other as related vectors. Knowledge learned about a specific word can also be applied to words related to it.

## 4.2 Word2Vec : Skip-Gram

Efficient dense representation can be done with well-trained word embedding model. Skip-gram is one of embedding models which predicts context with given word. Skip-gram

uses the idea that the closer the word is to sampling around the current word, the more relevant it will be to the current word. The concept of a window make makes context analysis possible. Given the input word in a continuous corpus, the learning proceeds with the word as the correct answer for the size of the window back and forth. All words are subject to input data and continuously proceed training.

More mathematically, the skip-gram model consists of a neural network model. V-dimension data will be used as input data. It also represents size of vocabulary. N is size of hidden layer. It indicates how many dimensions we will embed the word. The input is a one-hot encoded vector. Weight matrix $W_1$ between input layer and output layer has form of V×N matrix and is multiplied with input vector. Similar process is carried out with another weight matrix $W_2$ between hidden layer and output layer with size of N×V.

$$h = W^T x := v_{W_I}^T$$

· $v_{W_I}^T$ : the k-th row of W to h

$$u_j = v'^T_{wj} h$$

· $v'^T_{wj}$ : the j-th column of the matrix W'

Finally, the output layer has a dimension as many as the number of words again. This is because the output is a target word, so there are as many cases as the number of words This calculation is repeated as many times as the number of context words determined through window size. Finally, softmax function is used to change the prediction

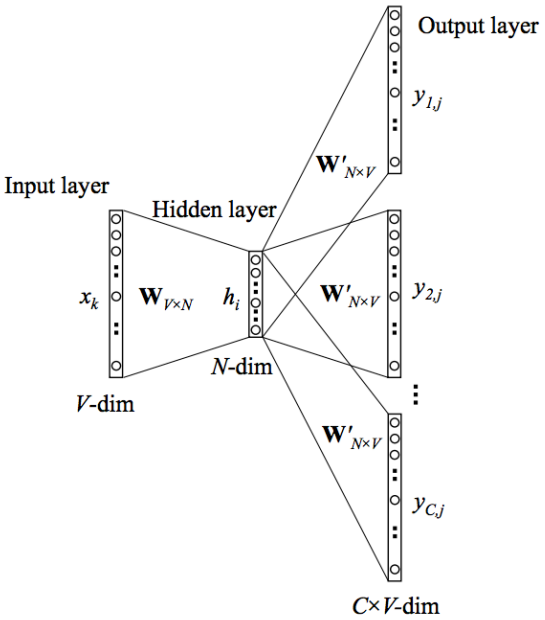



Figure 3. structure of skip-gram model

score to the probability value of each word. This is a method of making the score a probability in proportion to the score of each word. Through this method, the predicted score of each word is equal to or greater than 0, and when all are added, it changes to a probability value of 1.

$$p(w_{c,j} = w_{O,c} | w_I) = y_{c,j} = \frac{\exp(u_{c,j})}{\sum_{j'=1}^{V} \exp(u_{j'})}$$

· $w_{c,j}$ : j-th word on the c-th panel of the output layer

· $w_{O,c}$ : actual c-th word in the output context words

· $w_I$ : only input word

· $y_{c,j}$ : output of the j-th unit on the c-th panel of the output layer

· $u_{c,j}$ : net input of the j-th unit on the c-th panel of the output layer

Now, to calculate the loss function, we assume that each word is independent. In other words, we assume that the surrounding words are completely independent for the central word and calculate the conditional probabilistic function.

$$\mathrm{E} = -\log p(w_{O,1}, w_{O,2}, \cdots, w_{O,C}|w_I)$$
$$= -\log \prod_{c=1}^{C} \frac{\exp(u_{c,j_c^*})}{\sum_{j'=1}^{V} \exp(u_{j'})}$$

By chain rule, in order to minimize the value of loss function E, weight has to be updated as following.

· update $W_2$

$$v'^{(new)}_{wj} = v'^{(old)}_{wj} - \eta \cdot EI_j \cdot h$$

where $EI_j = \sum_{c=1}^{C} e_{c,j}$ , $e_{c,j} = y_{c,j} - t_{c,j}$

· update $W_1$

$$v^{(new)}_{wI} = v^{(old)}_{wI} - \eta \cdot EH^T$$

where $EH_i = \sum_{j=1}^{V} EI_j \cdot w'_{ij}$

# 5. Experiment

## 5.1 Dataset

Datasets are mxl file which is included in library. We analyze the songs of three musicians who were active in the three periods of Baroque, Classicism, and Romanticism. Following are piece number used in analysis.

· Johann Sebastian Bach (Baroque)
: bwv1.6 ~ bwv438, bwv846
· Ludwig van Beethoven (Classicism)
: opus18.no1, opus59.no1,2,3
· Robert Alexander Schumann (Romanticism)
: opus41.no1, opus48.no2

## 5.2 Experiment settings

**Hyperparameter**

· embedding dimension: 4
· learning rate: 1e-4
· window size: 2

Two of weight matrices are initialized by uniform distribution between -0.8 and 0.8.

· # of epochs: Bach(36), Beethoven(79), Schumann(75)

# 6. Results

## 6.1 Visualization of Data

First, the adjoining matrix created earlier was used to discover the pattern or trends of songs according to the era.

The higher the value of the matrix, the higher the association, the distribution of dots (high values, corelated notes) has different shapes in response to the trend of the times.

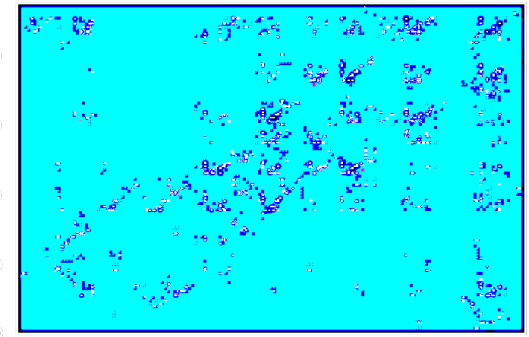


Figure 4-1. heatmap of adjacency matrix (Bach)

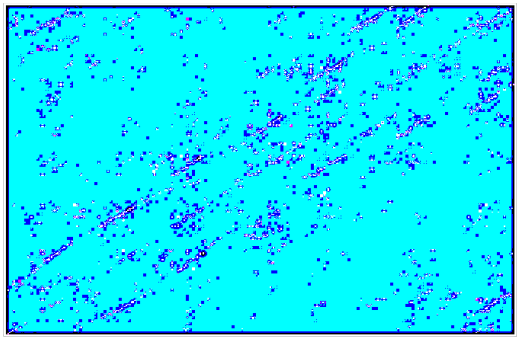


Figure 4-2. heatmap of adjacency matrix (Beethoven)

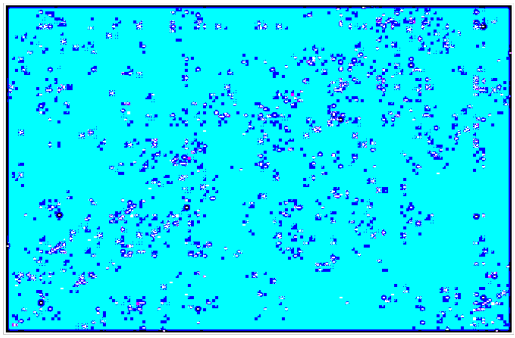


Figure 4-3. heatmap of adjacency matrix (Schumann)

The classical trend, which is characterized by formalization, has a clear regularity in the heat map, and dots are widely distributed in the romantic trend, which means notes are freely related.

## 6.2 Prediction

From each trained model, we predicted notes sequentially starting from ('C4', 1.0) that implies C4 tonic note located at starting point. When we predict notes with model, some of limitation existed by musical technique so some of manual work was required. The biggest reason why manual work was needed was that predictions for certain notes always appeared the same. We need to address this by choosing based on some degree of randomness.

Although I proceed composing somewhat manually, skip-gram guides to fine direction. Short verse of predicted music is attached with file.

## 6.3 Visualization of Embedded Matrix

We represent the dense representation made by each skip-gram model in four dimensions. Through the t-SNE technique, the weight matrix W1 in which embedded has been performed is visualized. Since there is no space, I will post only the plot of the Beethoven's piece.

And then we marked with color for certain indices. First, we marked indices which is located in regular beat (1, 2, 3).

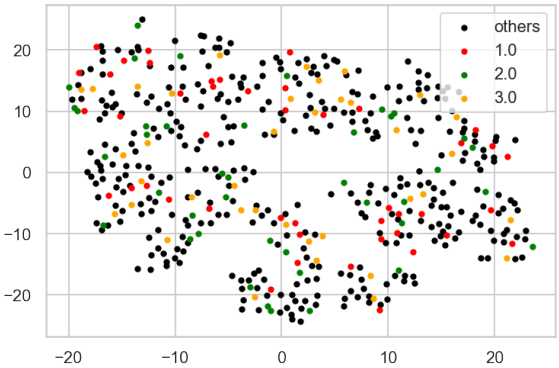



Figure 5-1. t-SNE plot of embedded matrix (w.r.t beat)

Secondly, we marked indices which has tonic pitches ('C4', 'E4', 'G4').

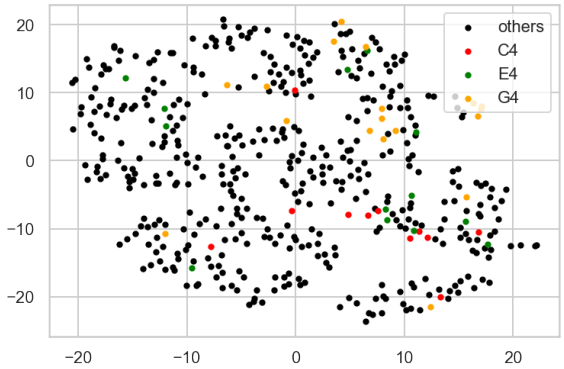



**Figure 5-2. t-SNE plot of embedded matrix (w.r.t pitch)**

## 7. Discussion

First, mxl data was imported and preprocessed to data in a form suitable for analysis. In order to check what pattern the given data show by era, an adjacent matrix between notes was created and visualized as a heat map. In addition, the skip-gram model was used to treat musical notes as corpus words, and a model for predicting context from a given input was created respectively by era. The model for each era was sequentially derived what results were predicted when the certain note was given. Finally, the weight matrix embedded in the skip-gram model was plotted by using t-SNE, and each point was colored by pitch and beat.

Since composing requires very complex technique, it was difficult to use only skip-gram model to predict and compose music. However, it suggests that if the skip-gram model is well processed, it can be an excellent foundation for automatic composition programs.