BOOSTING CLASSIFIERS ON TRANSFORMERS FOR DISCRIMINATING MUSIC CONTINUATIONS

Vane Wu **YuanLiang Dong** **Yu Hong** **Bin Wu** **Simon Lui**
Tencent Music Entertainment Group
{vanewu,gunterdong,yuuhong,benbinwu,nomislui}@tencent.com

ABSTRACT

This submission presents a model for MIREX 2020: Patterns of Prediction implicit task. The midi excerpts are encoded by both Music Transformer model and a statistical model. Then, different classification models are utilized in determining whether the continuation is the true continuation or foil. XGBoost algorithm is used for summarizing the predictions provided by those classifiers in order to get a more reliable classification result.

1. INTRODUCTION

Algorithmic composition is a hot topic in computer music and artificial intelligence research fields. However, it is still hard for computer to generate a natural-sounding song from a given theme. In most cases, people can easily to distinguish whether the excerpt is composed by a human composer or generated by an artificial intelligence system even if the prime theme is provided.

Some researchers deem that, the difficulty for generating a natural-sounding song, is due to the hardness for an algorithm to find the pattern of the music. If an algorithm is good at discovering the patterns of a given theme, it should also be good at making correct predictions for the continuation of that prime [1].

The implicit task of "patterns for prediction" in MIREX 2020 is to calculate the probability that the given excerpt is the true continuation of the provided prime. In order to predict the probability accurately, the designed algorithm should be good at finding the patterns of the prime and the continuation. This work is an attempt of that implicit task. We implement a classification model to calculate the probability, and the detailed architecture of our model will be described in the next section.

1.1 Related works

MIDI can be represented by event sequence. Finding the patterns of the sequential data have been studied by many researchers. Transformer model [2] is the most popular model in recent years, since it performed quite well in nature language processing (NLP) task. Music transformer model [3] is an improvement of the Transformer, and it is specially designed for algorithmic composition task. Therefore, we use it in our system as one of the MIDI data encoders, in order to extract the feature of the input.

Support vector machine (SVM) and neural network (NN) are famous and classical algorithms. The performances of those two algorithms are still good if the feature is not very high dimensional and the dataset is not very large. Therefore, we use it in our system to classify whether the given continuation is true continuation of the prime or not, after feature extraction stage.

XGBoost [4] is a popular gradient boosting framework in recent years. It can convert several weak learners to strong ones. In our system, it is used for summarizing the prediction provided by different classifiers.

2. CLASSIFICATION SYSTEM ARCHITECTURE

The details of our classification model designed for this task will be presented in this section.

2.1 Overall Architecture

Figure 1 shows the overall architecture of designed system. The MIDI is represented as an event sequence first, then a distribution or an encoding is extracted from this sequence. Next, several different models are trained to distinguish the true continuation and the foil by leveraging their encoding. Finally, XGBoost is trained for summarizing the prediction of those models.

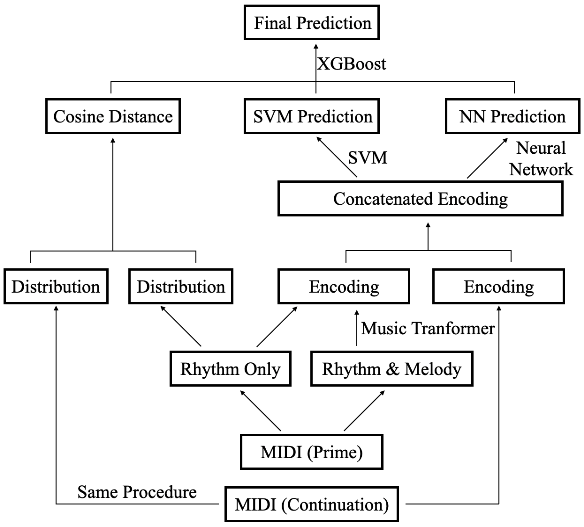

Figure 1. The overall architecture of our model.

2.2 MIDI Representation

According to our observations, the rhythm of the prime and the rhythm of the continuation are similar in most cases. Therefore, we extracted the rhythm information (i.e., the duration of each note) from the original MIDI data as another input, to better capture the rhythm pattern.

Moreover, we represent the MIDI data by two different ways, which is not distinguished in Figure 1. One is to treat the pitch event and the time-shift event separately, that is, each note is represented by three events: note-on, time-shift, and note-off. The other is to integrate the pitch event and the time-shift event, that is, each note is represented by only one event and the event collection is the Cartesian product of pitch events and time-shift events. In order to avoid the event collection being too large, we limit the pitch event to only 64 possibilities and the time-shift event to only 14 possibilities.

2.3 Encoders

We encode the event sequences by two methods: (1) calculating the distribution, (2) training a music transformer model. In practice, the transformer model is trained as a next note prediction model. That is, in training stage, the input of the transformer is an event sequence, while the output is the same sequence, but one-step advanced.

2.4 Classifiers

After encoding, we trained SVM and NN classifiers to predict whether the continuation is the true continuation or foil. The cosine distance between the rhythm distribution of the prime and the continuation, can also be used as a classifier, although it does not require training. Then, the XGBoost framework is trained for summarizing the prediction provided by different classifiers in order to get a more reliable classification result.

3. EXPERIMENTS

The datasets and parameters in training Music Transformer, SVM, NN, and XGBoost will be described in this section. The achieved accuracies before and after boosting will be also proposed.

3.1 Dataset

The “symbolic, monophonic, large” official dataset, which contains 10000 MIDI samples, is used for training and testing. The training set contains 9000 samples, which are randomly picked, while the test set contains the remaining 1000 samples.

3.2 Experimental Setup

Table 1 presents the major parameters we used in model training. Three values correspond to Rhythm only, Rhythm & Melody Independently, and Jointly model.

Model	Parameter	Value
	Sequence Length	128 / 512 / 256
Music Transformer	Model Dim.	16 / 64 / 64
	Number of Heads	3 / 6 / 6
	Learning Rate	0.001
	Solver	Adam
SVM	Kernel	RBF
Neural Network	Learning Rate	0.001
	Solver	Adam
XGBoost	Booster	Tree
	Gamma	0.1
	Lambda	2
	Max Depth	9
	Eta	0.007

Table 1. The major parameters in model training.

The hardware we used is: (1) GPU: Tesla V100, (2) CPU: Intel Xeon CPU, 2.5GHz, 36 cores. (3) Memory: 160GB.

3.3 Results

Table 2 presents the accuracy of each classifier, and the final accuracy we achieved after boosting. The accuracy we reported is a pairwise accuracy. That is, given a prime, if the predicted probability of the true continuations is larger than the foil, we treat it as a correct prediction.

Model	Accuracy
Cosine Distance Classifier	81.6%
Rhythm Only, SVM	95.5%
Rhythm Only, NN	94.4%
Rhythm & Melody Independently, SVM	94.7%
Rhythm & Melody Independently, NN (Not used in XGB since no improvement)	95.3%
Rhythm & Melody Jointly, SVM	94.9%
Rhythm & Melody Jointly, NN	94.1%
XGBoost (Final Model)	97.5%

Table 2. The accuracy we achieved in the test set.

4. REFERENCES

[1] Meredith, David. "COSIATEC and SIATECCompress: Pattern discovery by geometric compression." International society for music information retrieval conference. International Society for Music Information Retrieval, 2013.

[2] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

[3] Huang, Cheng-Zhi Anna, et al. "Music transformer: Generating music with long-term structure." International Conference on Learning Representations. 2018.

[4] Chen, Tianqi, et al. "Xgboost: extreme gradient boosting." R package version 0.4-2 (2015): 1-4.