Modeling Self-Repetition in Music Generation using Generative Adversarial Networks

Harsh Jhamtani [1] Taylor Berg-Kirkpatrick [1 2]

Abstract

In this paper we propose a generative model for music generation focusing on self-repetition. We use a Generative Adversarial Network formulation to learn a model which can generate compositions with long term repetition structures similar to those found in training data. We propose to represent self-repetition in a composition using a self-similarity matrix constructed by computing similarity between pairs of measures. To avoid optimization issues due to discrete nature of notes in musical compositions, and to provide more flexibility in identifying similarity between measure pairs, we propose to encode measures into low dimensional measure embeddings – effectively lifting the discrete observations to a continuous space. Our model reasons about generating structured sequences directly in this lifted space. Preliminary experiments show promising results from our proposed method.

1. Introduction

Musical compositions often demonstrate repetitions (Pareyon, 2011; Walder & Kim, 2018), in terms of patterns related to rhythm, pitch, and other musical properties. Some prior works focus on modeling long term structures in music generation (Eck & Schmidhuber, 2002; Huang et al., 2018; Roberts et al., 2018). However, there are only few works on explicitly representing self-repetition (Walder & Kim, 2018). In this paper, we propose **SSMGAN** (Figure 1) - a generative adversarial network (Goodfellow et al., 2014) for learning a neural model to generate monophonic compositions with rich self-repetition structures by feeding a measure-level **s**elf-**s**imilarity **m**atrix representation to a convolutional discriminator, which can be more informative than taking localized decisions with self-attention. Instead of explicitly defining the notion of similarity between two measures, we propose to encode a measure - a sequence of notes - into a continuous representation, and compute a self-similarity matrix using pair-wise cosine similarity of measure representations/embeddings in the composition.

Prior works have often chosen to characterize repetition in terms of rhythmic or other manually defined musical properties, and edit distances at note level (Walder & Kim, 2018). However, musical similarity might go beyond such simple formulations (Flexer et al., 2006; Prockup et al., 2015). By representing measures in a continuous space, our model can learn more complex notions of similarities between measure. Additionally, we feed a self-similarity matrix to a discriminator D_S, which learns to identify repetition structures in existing compositions using multiple layers of 2D convolution neural networks. Moreover, since we represent measures in a continuous space, loss from D_S is fully differentiable with respect to the measure representations.

2. Methodology

Measure and Self-repetition representations: We propose to encode a measure N_i consisting of a sequence of notes $N_i^1, N_i^2, .., N_i^{|N_i|}$ to a low dimensional embedding M_i. We train a variational auto-encoder (Kingma & Welling, 2013) at measure level using a LSTM encoder, denoting the last hidden state of encoder as corresponding measure embedding M_i. However, instead of having a decoder operate on each measure embedding individually, we propose a decoder which has access to the last hidden state of previous measure's decoder (Figure 1). The proposed decoder can lead to smoother transition across measure boundaries. We define **self-similarity matrix $\mathbf{S_M}$** $\in R^{T*T}$ such that S_{ij} is the cosine similarity score between pair of measures N_i and N_j, while T is the number of measures in the composition.

GAN formulation Generative Adversarial Networks (GANs) employ two types of networks - *generator* and *discriminator*, such that discriminator is trained to identify generated examples from training examples, and the generator is trained to *fool* the discriminator.

Neural Generation Model We first sample a sequence of

[1] School of Computer Science, Carnegie Mellon University, USA [2] University of California San Diego, USA. Correspondence to: Harsh Jhamtani <jharsh@cmu.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

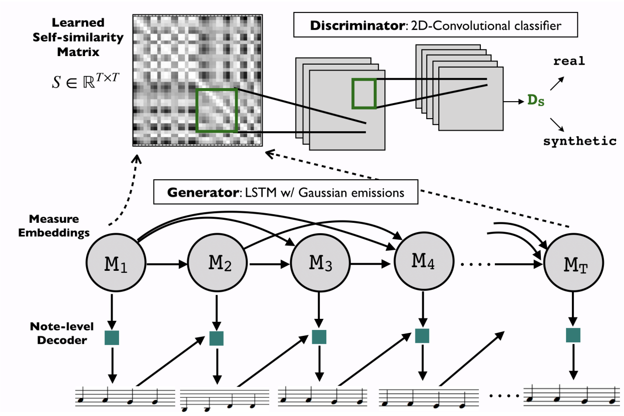
Learned Self-similarity Matrix
$S \in \mathbb{R}^{T\times T}$
Discriminator: 2D-Convolutional classifier
real
D_S
synthetic
Generator: LSTM w/ Gaussian emissions
Measure Embeddings
M_1
M_2
M_3
M_4
M_T
Note-level Decoder

Figure 1. SSMGAN model: We sample a sequence of measure embeddings from the generator, which are decoded to obtain sequences of notes. One of the discriminators operates on a self-similarity matrix obtained via cosine similarity between pairs of measure embeddings. The proposed architecture encourages the model to identify useful notions of similarity between measures, and identify overall self-repetition patterns from data.

measure embeddings $M_1, .., M_T$ such that M_i depends on all $M_{<i}$ (Figure 1). We sample M_i from a Gaussian distribution whose mean and variance are computed by feeding the output h_i of the LSTM to two feed-forward networks. We use re-sampling trick to train the model end-to-end. Thereafter, the decoder (same as the one used during training measure representations) decodes the sampled sequence of measure embeddings into the final sequences of notes.

Discriminators: We employ two discriminators operating on 1) Self-similarity matrix 2) Sequence of measures.

$\mathbf{D_S}$: The discriminator $\mathbf{D_S}(S)$ uses a multi-layer convolutional encoder to encode a self similarity matrix S and is trained to distinguish S of a generated composition from that of a training data composition.

$\mathbf{D_L}$: We consider a convolutional neural network discriminator $\mathbf{D_L}$ which looks at windows of K measure embeddings at a time. We encode the sequence using a LSTM, followed by a linear layer and a sigmoid to model this binary classification discriminator. We experiment with multiple values of K.

Training: The generator and the discriminators are trained simultaneously. We run the encoder on training data compositions to obtain a sequence of *real* measure embeddings, while we sample from the generator to get *synthetic* compositions. We note that training the full model end-to-end can become problematic as the measure encoder can work with the generator to *fool* the structure discriminator at the cost of generating good measure representations. So instead we pre-train and then freeze the parameters of the measure encoder-decoder. Thereafter, we use GAN framework to train the LSTM Gaussian model while keeping the measure encoder decoder fixed.

3. Experiments and Results

We work with Nottingham dataset (**GOLD**) (Shlien; Boulanger-Lewandowski et al., 2012) which is a collection of 1200 British and American folk tunes, with over 7 hours of music with a total of over 176K notes. We identify the measure boundaries in every composition. Following prior works, we perform listening tests with human annotators on Amazon Mechanical Turk. Annotators rate the generated samples on overall quality **O** on a 1-5 Likert scale, (with 5 being the most favorable score), and a binary yes/no question about presence of repetition **R** (Table 1).

Model	R (% yes)	Overall (O)
SSMGAN	63%	4.28
SSMGAN (-D_S)	26%	3.84
NOTE	27%	4.02
GOLD	64%	4.64

Table 1. Human evaluation results with 108 samples of each method on 1) self-repetition (R) 2) overall musical quality for our method SSMGAN, a note level LSTM baseline (NOTE), and samples from Nottingham data (GOLD). SSMGAN (-D_S) denotes our method without D_S discriminator.

Measure Embeddings and Self-Similarity: We use LSTM cells with hidden size of 128 as measure encoder and decoder, with note embedding size of 128. As discussed, during pre-training phase we only train measure encoder-decoder, and observe 1.137 note-level perplexity on test split of the Nottingham data. Additionally, we observe similar measures are close-by in embedding space. (Some relevant visualizations in Appendix B). We observe that our model learns to generate sequences with rich repetition structures (Some relevant visualizations in Appendix A). Moreover, we observe that cosine similarity between pairs of measure embeddings is correlated with a note level edit distance of pitch as well as rhythm between measures (Pearson correlation coefficients of 0.40 and 0.35 respectively), demonstrating that proposed self-similarity matrices encode repetition in terms of meaningful musical properties.

Other Related Works: Dong et al. (2018) and Yang et al. (2017) propose GAN based methods for music generation, with latter using a 2D convolutional discriminator on sequence of generated bars. Widmer et al (2018) use a convolutional restricted boltzmann machine to generate music while imposing a given repetition structure of a piece. In contrast to such earlier works, we have proposed to encode measures in a low dimensional embeddings space, and use a discriminator on self-similarity matrix - enabling our model to automatically learn useful notions of similarity between measures, and identify meaningful self-repetition patterns from data.

References

Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. *arXiv preprint arXiv:1206.6392*, 2012.

Dong, H.-W., Hsiao, W.-Y., Yang, L.-C., and Yang, Y.-H. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Eck, D. and Schmidhuber, J. A first look at music composition using lstm recurrent neural networks. *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale*, 103, 2002.

Flexer, A., Gouyon, F., Dixon, S., and Widmer, G. Probabilistic combination of features for music classification. In *ISMIR*, pp. 111–114, 2006.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Hawthorne, C., Dai, A. M., Hoffman, M. D., and Eck, D. An improved relative self-attention mechanism for transformer with application to music generation. *arXiv preprint arXiv:1809.04281*, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Pareyon, G. *On Musical Self-Similarity: Intersemiosis as Synecdoche and Analogy*. Gabriel Pareyon, 2011.

Prockup, M., Ehmann, A. F., Gouyon, F., Schmidt, E. M., and Kim, Y. E. Modeling musical rhythmatscale with the music genome project. In *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 1–5. IEEE, 2015.

Roberts, A., Engel, J., Raffel, C., Hawthorne, C., and Eck, D. A hierarchical latent vector model for learning long-term structure in music. In *International Conference on Machine Learning*, pp. 4361–4370, 2018.

Shlien, S. Nottingham dataset. http://ifdo.ca/œseymour/nottingham/nottingham.html.

Walder, C. and Kim, D. Neural dynamic programming for musical self similarity. In *International Conference on Machine Learning*, pp. 5092–5100, 2018.

Widmer, G., Grachten, M., and Lattner, S. Imposing higher-level structure in polyphonic music generation using convolutional restricted boltzmann machines and constraints. *Journal of Creative Music Systems*, 2(2), 2018.

Yang, L.-C., Chou, S.-Y., and Yang, Y.-H. Midinet: A convolutional generative adversarial network for symbolic-domain music generation. *ISMIR*, 2017.

Appendix

A. Visualizing self-similarity matrices

Figure 2 visualizes SSM (self-similarity matrices) of some generated compositions.

B. Visualizing measure embeddings

Figure 3 visualizes sequence of measure embeddings for a composition. Many of the measure embeddings seem to be falling into groups. For example, M_4, M_5 and M_8 seem very close in the t-SNE space. In fact on inspecting the corresponding sequence of notes, we observe that N_4, N_5 and N_8 are indeed very similar.
Notation: Sequence of notes where each note is represented by a tuple of key and quarter length).
N_4 = [(D5, 0.5), (C5, 0.25), (B4, 0.25), (A4, 0.25), (B4, 0.25), (C5, 0.25), (B4, 0.25)]
N_5 = [(C5, 0.25), (D5, 0.25), (E5, 0.5), (G4, 0.25), (F4, 0.25), (E4, 0.5)]
N_8 = [(D5, 0.5), (C5, 0.25), (B4, 0.25), (A4, 0.25), (B4, 0.25), (C5, 0.5)]

C. Generated Samples

Some generated musical composition samples can be found at https://drive.google.com/drive/folders/1TlOrbYAm7vGUvRrxa-uiH17bP-4N4e9z?usp=sharing.

D. Pre-training Measure Embeddings

We pre-train our measure embeddings using a variational autoencoder (Figure 4).

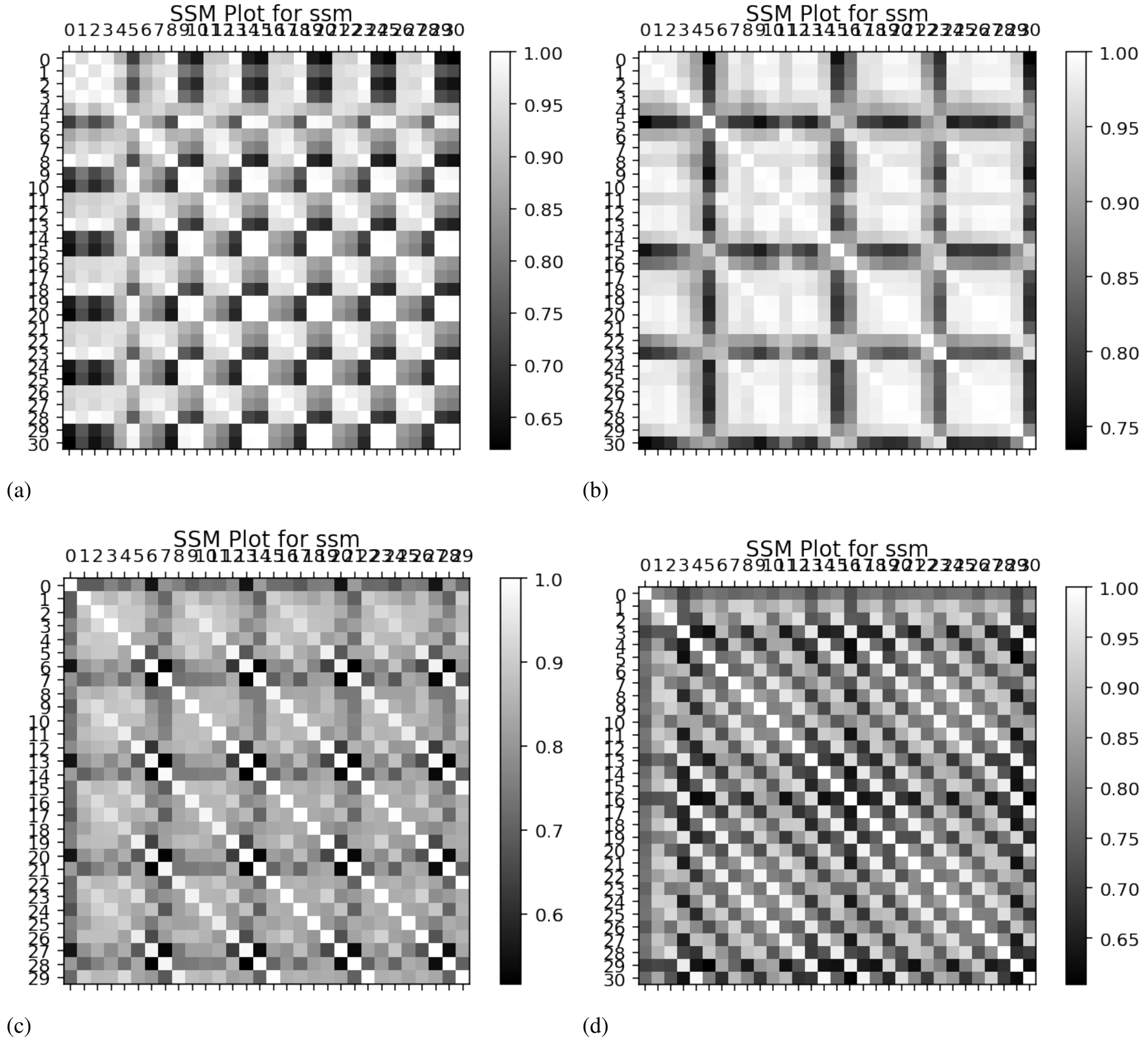

Figure 2. **APPENDIX A:** SSM (Self-similarity matrix) visualizations for some generated music compositions. Rows and columns are numbered with measure numbers starting with 0. Higher values represents higher similarity between pair of corresponding measures.

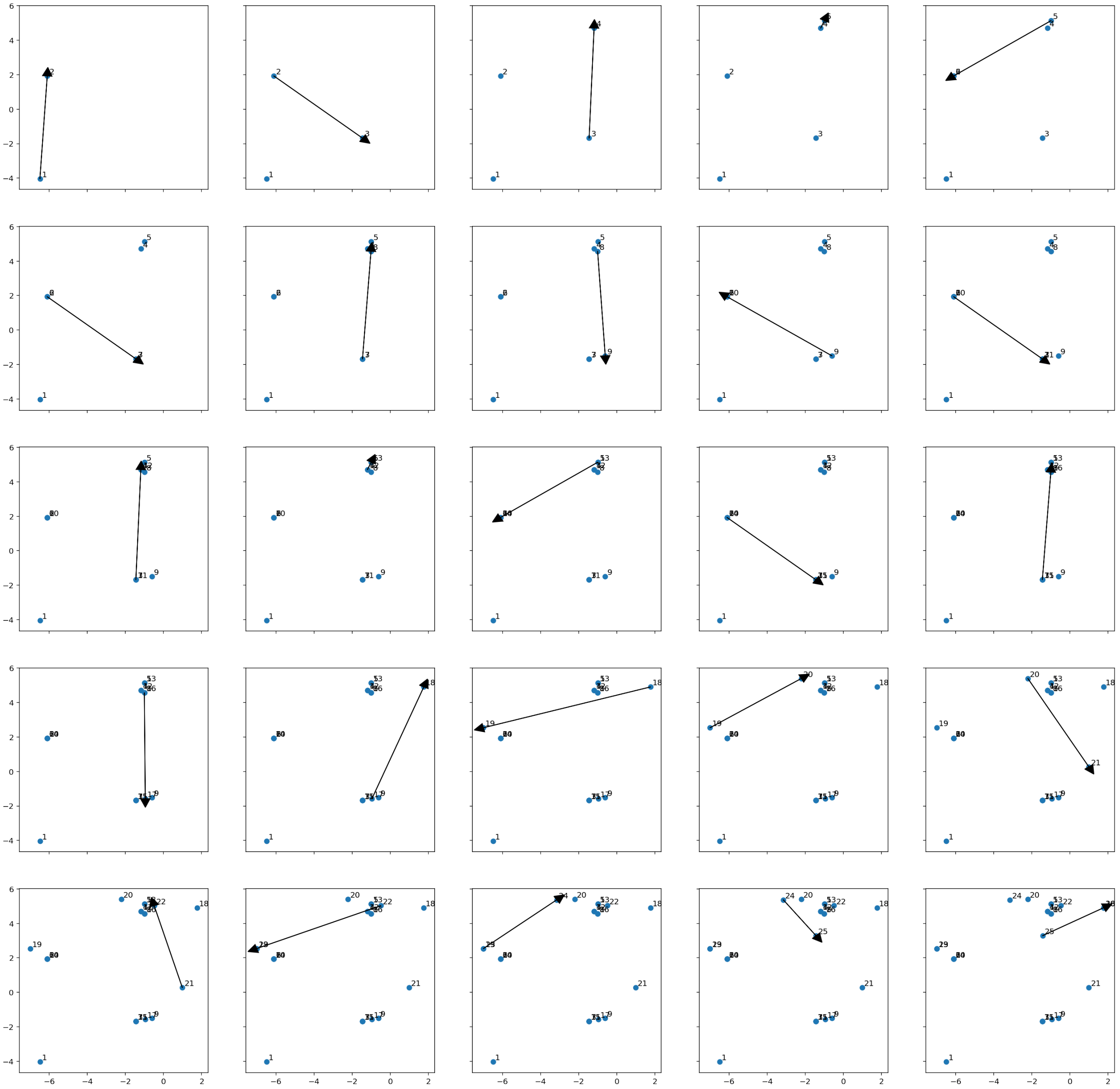

Figure 3. **APPENDIX B:** t-SNE visualization of a sequence of 25 measure embeddings in a Nottingham composition (Subplots are arranged in row-major style). Groups of measures can be observed in t-SNE space which is in sync with the observation that measures are repeated with small changes.

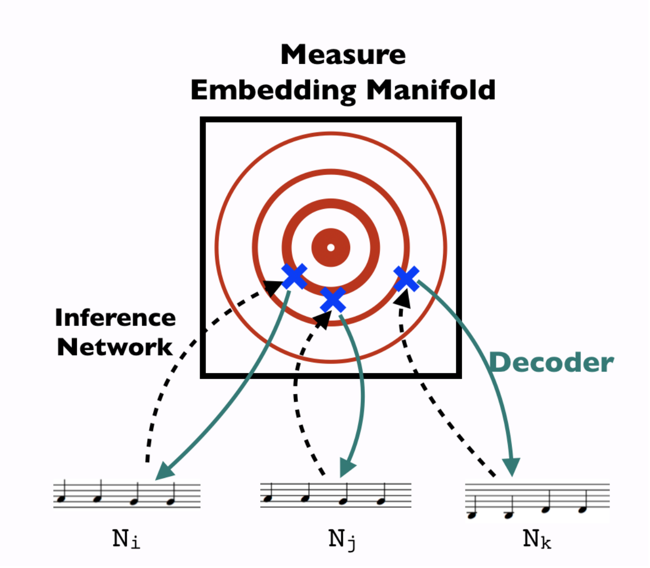

Figure 4. **APPENDIX D:** We pre-train our measure embeddings using a variational autoencoder.