COMPARATIVE RNN PERFORMANCE ACROSS MUSICAL GENRES AND INSTRUMENT CLUSTERS 2020

Agustín Krebs
Pontificia Universidad Católica de Chile
akrebs2@uc.cl

Alexander Rusnak
Ecole Polytechnique Federale de Lausanne
alexander.rusnak@epfl.ch

Axel Sjöberg
Lund University
ax3817sj-s@student.lu.se

ABSTRACT

In this work we use a recurrent neural network (RNN) trained on modified MIDI file data to generate four separate instrument tracks for three genres; pop-rock, ballad and house music. The MIDI files comes from the million song dataset and the genre tags are provided by the echo-nest lab. The data representation was done by first by mapping the instruments to a 4-families classification and then using a quantized-time encoding approach. The genre tags were filtered using a normalized frequency threshold criteria. Feeding an RNN with an LSTM cell we trained 16 models, 1 for each instrument-genre pair. Using these models, distinctions in loss on the test set between instruments, genres and features were identified and discussed. We found that the melodic instruments carried more complexity in the ballad and pop rock genres than the house genre, while the more rhythmic instruments like drums and bass were more difficult to learn for the house genre. The paper is concluded with some analysis on weaknesses in the method and suggestions on how these can be improved in future research.

1. INTRODUCTION

One of the fundamental cornerstones in categorizing, organizing and describing music is genre. Borrowed from French, where it literally translates to - "a kind", musical genres are used to relate a song to a larger group where the members share typical characteristics. The characteristics that form the basis for differentiating and describing genre are often related to the instrumentation, the harmonic content and the rhythmic structure. The task of articulating precisely the difference between genres is a complex one due to the lack of formal definitions and the soft boundaries between neighboring styles. Historically, genre classification have been made by manual annotation, but with the surge of available songs in digital format, there has been an explosion in the amount of available data needed for computational analysis of genre. However, in 2014, Sturm [6] showed that many at the time state of the art Music Information Retrival (MIR) Systems, had misleading figures of merit (FoM) as the systems often were relying on characteristics in the dataset confused with the ground-truth.

In this research we use a recurrent neural network architecture with LSTM cells to train 16 models on modified MIDI file data, each to generate a specific instrument (out of piano, guitar, drums, and bass) track for generally similar genres of music: ballads, pop rock, and house music. With this method our ambition is to minimize our models reliance on characteristics confounded by predefined conceptions of ground truth from musicology and to use the observations of the model to elucidate some of the characteristics that have been less explored in defining genres and instrument differences.

2. RELATED WORK

One of the earlier works on the subject was done in 2002 by Eck and Schmidhuber where they demonstrated that a RNN can capture not only the local structure of a melody but also the long-term structure of a musical style. [2]

In 2017 Kotecha and Young, presented a LSTM approach to composing polyphonic music. [4] The authors showed that the approach was successful and moreover suggested that future models include genetic algorithms. The same year Zhou et al [9] presented Bandnet; a Beatles-style composition machine based on a RNN that had been trained using MIDI-files. Using Bandnet they were effective in automatically generating music in the style of Beatles. This implementation was similar to ours in the sense that it combined multiple instrument tracks composing concurrently, however they choose to pass vectors representing RNN cell state between instruments while we opted to give each layer access to the input data of every instrument.

Chung Pui Tang et al [7] showed in 2018 that a LSTM model that was trained on the GZTAN dataset [8] was effective in classifying musical genres.

3. PREPROCESSING

3.1 Datasets

For this work, we used the data from the project Million Song Dataset [1] and the Lakh MIDI Dataset v0.1 [5] . In particular, we used the subset called "lmd aligned" from Lakh MIDI Dataset, which consisted of 45,129 songs aligned to the 7digital preview MP3s. As you can see in Figure 1, for each song there were available a variable number of MIDI representations. Also, for each song, it

Attribution: Agustín Krebs, Alexander Rusnak, Axel Sjöberg. "Comparative RNN Performance Across Musical Genres and Instrument Clusters 2020", 21st International Society for Music Information Retrieval Conference, Montréal, Canada, 2020.

was possible to obtain its artist. With the artist information, it was then possible to obtain a variable number of music genres associated to it. These genre tags came from the echo-nest lab and have been collected using automatic annotation based on web-scraping.

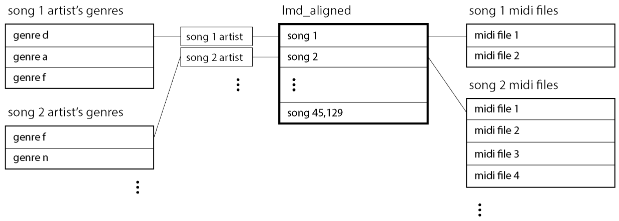


Figure 1. Dataset schema

3.2 Genre Selection

The genre tags available for all the artists are unsuitably large and filled with noise. The following information is contained in the genre tags:

- A list containing all of the unique genres the artist have been tagged with
- A list where the entries are the normalized frequency at which an artist have been tagged with a unique genre, where the normalizing factor was the frequency of the most frequent genre

Thus for every artist there is one list containing all the unique genres whom they have been tagged with, together with the normalized frequency number in the range [0,1]. The mean number of unique genres an artist have been tagged with was 11.59

By dropping all of the genres with a normalized frequency smaller than 0.9, a reduced unique genre lists was obtained for every artist. This was done in order to reduce the irrelevant genres which the artist by accident might have been tagged with and to only get the most relevant tags. After this reduction, each artist was on average tagged with 3.5 genres.

Each of the songs in our dataset was paired with their artist genre tags. In Figure 2, the x-axis shows the most frequent genres in our dataset and the y-axis shows the percentage of songs tagged by each of these genres. Note that one song can be tagged by several genres, i.e every Radiohead song is tagged by pop-rock and hard-rock.

Figure 3 shows the Jaccard similarity, defined in equation 1, for some of the most popular genres. As can be seen in the figure, there are many songs that are tagged with the same genres. Some genres for which this phenomena is very pronounced are progressive house, hard house and trance. In order to increase the amount of data for the genres which the models will be trained on, the following genres were merged: Progressive house, hard house, deep house, trance and hard trance.

$$J(A,B) = \frac{A \cap B}{A \cup B} \tag{1}$$

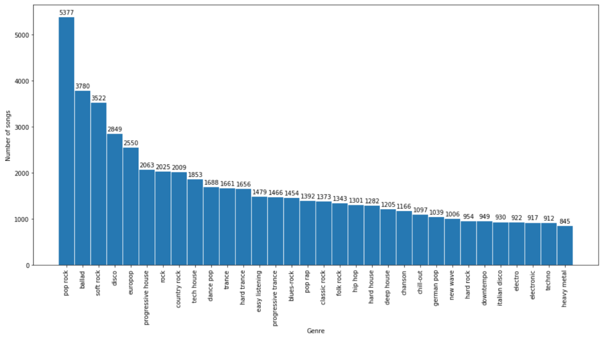
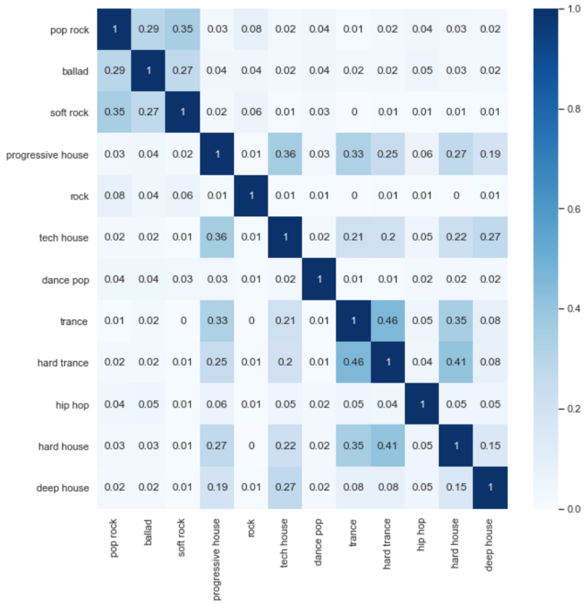


Figure 2. Genre tag frequency for the most popular genres

In addition to the house music genre, we used pop rock and ballads as the genres to train the models. The reason for choosing pop rock and ballads as genres to train the models was that they are the two most frequent genres. The reason for choosing the merged house genre was that the house genre had a large amount of data in addition to being rather uncorrelated to the other two genres.



	pop rock	ballad	soft rock	progressive house	rock	tech house	dance pop	trance	hard trance	hip hop	hard house	deep house
pop rock	1	0.29	0.35	0.03	0.08	0.02	0.04	0.01	0.02	0.04	0.03	0.02
ballad	0.29	1	0.27	0.04	0.04	0.02	0.04	0.02	0.02	0.05	0.03	0.02
soft rock	0.35	0.27	1	0.02	0.06	0.01	0.03	0	0.01	0.01	0.01	0.01
progressive house	0.03	0.04	0.02	1	0.01	0.36	0.03	0.33	0.25	0.06	0.27	0.19
rock	0.08	0.04	0.06	0.01	1	0.01	0.01	0	0.01	0.01	0	0.01
tech house	0.02	0.02	0.01	0.36	0.01	1	0.02	0.21	0.2	0.05	0.22	0.27
dance pop	0.04	0.04	0.03	0.03	0.01	0.02	1	0.01	0.01	0.02	0.02	0.02
trance	0.01	0.02	0	0.33	0	0.21	0.01	1	0.46	0.05	0.35	0.08
hard trance	0.02	0.02	0.01	0.25	0.01	0.2	0.01	0.46	1	0.04	0.41	0.08
hip hop	0.04	0.05	0.01	0.06	0.01	0.05	0.02	0.05	0.04	1	0.05	0.05
hard house	0.03	0.03	0.01	0.27	0	0.22	0.02	0.35	0.41	0.05	1	0.15
deep house	0.02	0.02	0.01	0.19	0.01	0.27	0.02	0.08	0.08	0.05	0.15	1

Figure 3. Jaccard similarity for the most popular genres

Table 1, shows the Jaccard similarity between the genres. The Jaccard similarity between ballad and pop rock is rather high. This in itself is not something that is wrong, e.g plenty of the songs by bands like Journey and Foreigner could be classified as being both ballads and pop rock. The problem arises when there is a song by an artists in the dataset that is not in their predominant genre style (e.g. if one artist that mainly make ballads and then experiment and create a hip/hop or metal song this song will be classified as a ballad). Obscurities like these are however assumed to be very limited and they are therefore treated as acceptable noise.

	pop rock	ballad	house
pop rock	1.00	0.29	0.05
ballad	0.29	1.00	0.06
house	0.05	0.06	1.00

Table 1. Jaccard Index for the genres used to train the models

3.3 Instrument Merging

The next step in the preprocessing pipeline focused on separating the songs into instrument tracks. To understand this process, is important to acknowledge the fact that a MIDI file is able to encode separately 128 pitch-instruments (also encoded as non-drum instruments). Having that said, the distribution of the instruments across the dataset was undoubtedly unequal, as is it shown in Figure 4. Considering this scenario, we realized that we were facing a dimensionality problem, since the amount of available MIDI files will not be enough for the RNN to generalize across this amount of different instruments.

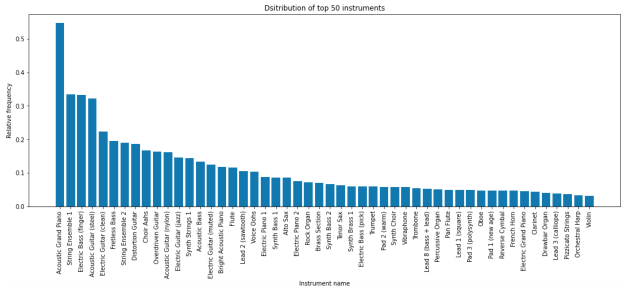


Figure 4. Distribution of the top 50 instruments

Taking these issues into account, we decided to make an abstraction step to transform the "instrument space" into a smaller one. Specifically, we mapped the original instruments into a 4-families classification: drums, pianos, basses, and guitars. On that premise, each instrument of a song was mapped to either one of the 4 families, or none of them (basically, removing that instrument from the MIDI file).

The two main questions that arise from the previous paragraph are the following: (1) why to choose those 4 specific instrument families? And (2) given those 4 families, how can you choose to which family is going to be assigned each of the 128 instruments?

Regarding the first question, we chose those specific 4 families of instruments because of their broadly well known usage. It is almost "common knowledge" that a western music band is conformed by a percussion instrument (usually drums), a piano, a bass and a guitar (also, it usually includes the voice lead). Because of this western universal cultural factor, we decided for this to be our prior instrument families.

Nevertheless, we still needed empirical data to validate this idea and to answer question (2). Because of that, we decided to perform a instrument hierarchical clustering in our dataset, in which we used pitch distributions to measure similarities between all the 128 pitch-instruments. We took each of the 128 possible pitches as a different feature or independent variable, and measured how present (weighted by note duration) was each of them across all the MIDI files of the dataset, for each pitch-instrument. With that, we performed a hierarchical clustering using euclidean distance, obtaining the cluster assignation shown in Figure 5. It is important to remark that we couldn't make the same analysis for drum instruments, since they don't have pitches differentiation. For simplicity, we decided to consolidate that group right away, since they come already separated in a MIDI file as a "drum instrument".

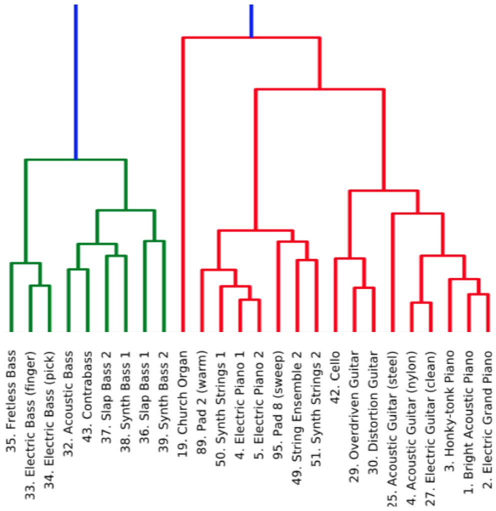
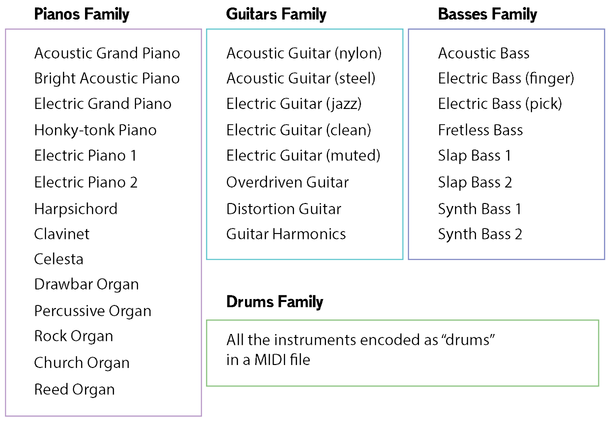


Figure 5. Snapshot of Instruments Clustering's Dendrogram

As you can see from the dendrogram (Figure 5), the bass family was really a defined group (they did not merge with other cluster until the final iterations of the algorithm). That was an interesting finding, since it matched perfectly with our prior intuitions or knowledge. However, we didn't find the same on the two remaining families (guitar and piano). As it is noticeable in Figure 5, guitars and pianos have mainly the same pitch distribution, since the algorithm merges them in the same cluster in a really early stage. Acknowledging the fact that this opens several more questions and different treatments for future research, we decided to keep both instrument families separate to avoid noise.. With all the previous said, we merged our prior knowledge with this clustering to arrive to the final instrument mapping shown in Figure 6.

Finally, we used this specific mapping to merge all notes from the same family into a single track. Even though a different technique of merging could have been applied (e.g. taking a random instrument per family instead of merging them all), we decided to use this one to be able to represent the instruments families "as a whole", instead of random subsamples of it. It is important to remark that all MIDIs that didn't have at least one instrument per family were removed from the dataset. This was done to avoid feeding the model with empty instrument family tracks.

In Figure 7 one can see the distribution of songs per genre after these 2 stages of preprocessing and filtering.



Pianos Family

Acoustic Grand Piano
Bright Acoustic Piano
Electric Grand Piano
Honky-tonk Piano
Electric Piano 1
Electric Piano 2
Harpsichord
Clavinet
Celesta
Drawbar Organ
Percussive Organ
Rock Organ
Church Organ
Reed Organ

Guitars Family

Acoustic Guitar (nylon)
Acoustic Guitar (steel)
Electric Guitar (jazz)
Electric Guitar (clean)
Electric Guitar (muted)
Overdriven Guitar
Distortion Guitar
Guitar Harmonics

Basses Family

Acoustic Bass
Electric Bass (finger)
Electric Bass (pick)
Fretless Bass
Slap Bass 1
Slap Bass 2
Synth Bass 1
Synth Bass 2

Drums Family

All the instruments encoded as "drums" in a MIDI file

Figure 6. Final Instrument Conversion

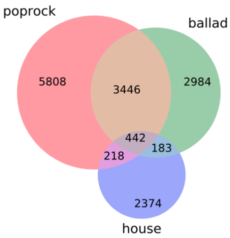


Figure 7. Genres Venn Diagram after preprocessing

3.4 Encoding

For the songs' encoding, we encountered the problem of how to translate the tracks information as an input to our model, specially regarding the time component. Giving the existing literature and our neural network, we chose to quantize our input into a discrete time representation. To better understand our encoding, it is helpful to look at Figure 8, a visual illustration of this process. On the left of that Figure, there is representation of a single track encoded in MIDI file (in our case, the track of one instrument's family). The track consists of an arrange of notes, sorted by their onset time. Each note has four basic features: pitch number (from 1 to 128), duration (in seconds), velocity (from 1 to 128), and onset instant of time (in seconds). It is easy to visualize this as "a piano roll" with a temporal component, in which the different notes land.

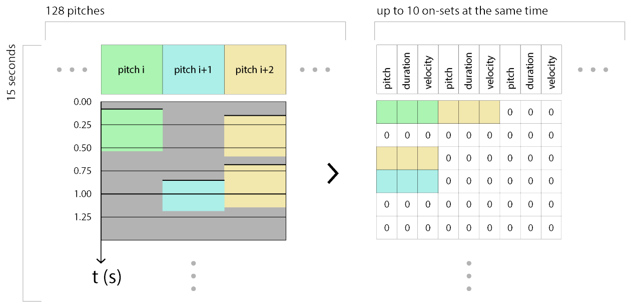


Figure 8. Songs Encoding Diagram

We used time steps of 0.25 seconds, which best preserved the input data without becoming too sparse. Given this, the encoding places each note in a time step if the onset of the note occurs in that interval. For every note which occurs in a given time step the pitch, duration, and velocity are encoded. Therefore, each row in this encoded song corresponds to a certain time step, where the columns represent groupings the 3 features of each note at that time step. We limited the total potential notes to 10 per time step, in order to once again limit the sparsity of the data. This was more than enough for most songs, but since our objective was to use these models to explore the data rather than create the most interesting compositions, we tried to preserve as much as the original data as possible. The columns are filled from left to right, excluding all notes that exceeded the aforementioned limit.

This encoding its capable of preserve intact 3 of the 4 features of a MIDI note (pitch, duration and velocity), and keeping an approximate version of the 4th one (onset time). Therefore, it was precise enough for the aims of this project.

4. CREATING THE MODEL

4.1 Explanation of a RNN

Artificial neural networks rely on a series of 'neurons' or units, each with an accompanying weight and bias, to transform input data into an accurate prediction of a piece of target data. The loss between the output of the neural network model and the target data is then calculated, and the weights and biases are adjusted to make the model more accurate in its predictions using a process called backpropagation. In general, neural networks have become incredibly popular in computer science in the last 10 years as research into their application exploded partially because of an increase in available computing power, advances in parallel computing, and large scale datasets becoming more prominent. Variations of neural networks have been used for processes as diverse as: computer vision, text processing, playing video games, financial instrument trading, and of course music analysis and generation.

We specifically chose a recurrent neural network because of its unique nature designed to focus on sequential data. Unlike a traditional network, the recurrent units of this type each have an output at each time step and a separate interior state prediction that is passed between the cells of each time step. We picked the very popular long-short term memory cell over several other options because of its suitability to handling larger and more sparse data rather than a cell like GRU which compresses the data substantially. In light of prior success in this field using RNNs, we think our choice was a good one.

4.2 Architecture

Our model architecture features an embedding layer with a shape of 200x512x60, 2048 recurrent units using an LSTM cell, and a dense output layer of 200x1. One batch of input (60 rows of the prior explained encoding method) is the equivalent of 15 seconds of music, and at each step the model predicts its expectation of the next 15 second snippet of the song. At each step the model receives the concatenated encoding of all four instruments from a given song in a given genre, but the loss is only determined in relation to one instrument during training. In this way, each model learns to optimize to output only one instrument. We used a learning rate of 0.005 with an adam optimizer [3] and sparse categorical cross entropy as our loss function.

With this model architecture defined, we trained 16 models (1 for each instrument for each genre) across the training set for four epochs each. The models were able to achieve relatively low loss values across all genres, partially due to how sparse the training and test data are. We were able to observe some broad distinctions in loss on the test set between various instruments, genres, and features.

5. RESULTS

Our initial finding was that the models for the pop rock genre were the most effective at generating convincing predictions. These models achieved a mean loss of 0.71407974, 0.677494 for piano, 1.224116 for guitar, 0.6553549 for drums, and 0.29935387 for bass. As you can see from figures 9 and 10, ballad genre loss was also low across the board though slightly higher at a global mean of 0.7616296. These two genres were very similar, with most loss occurring in the pitch features for the melodic instruments (fig. 11). This can be broadly attributed to arrangement complexity because house music had a higher note incidence rate than either pop rock or ballad. This finding is supported by figure 13.

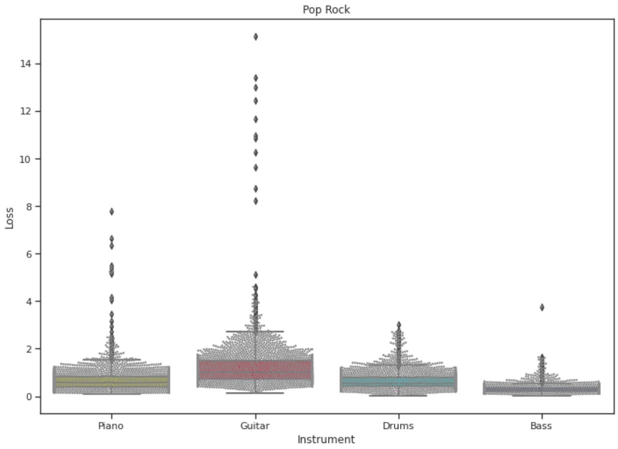


Figure 9. Pop Rock Genre Loss by Instrument

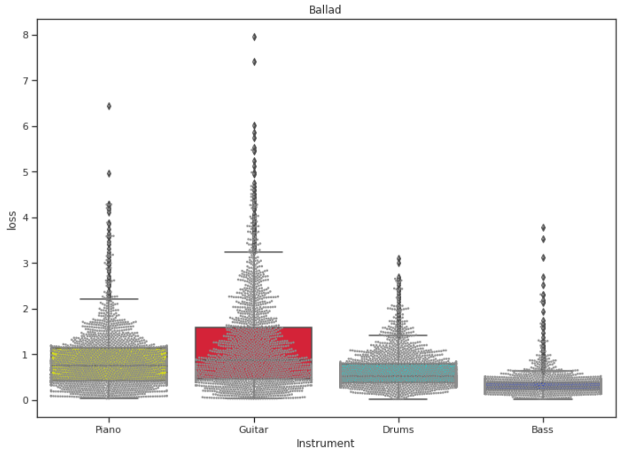


Figure 10. Ballad Genre Loss by Instrument

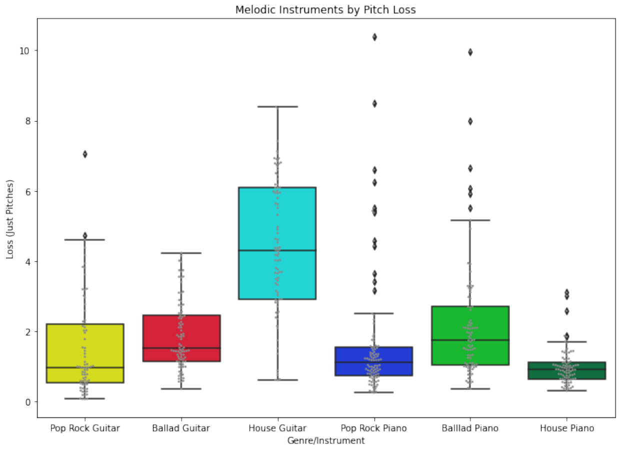


Figure 11. Pitch loss of Melodic Instruments (Piano, Guitar) by Genre

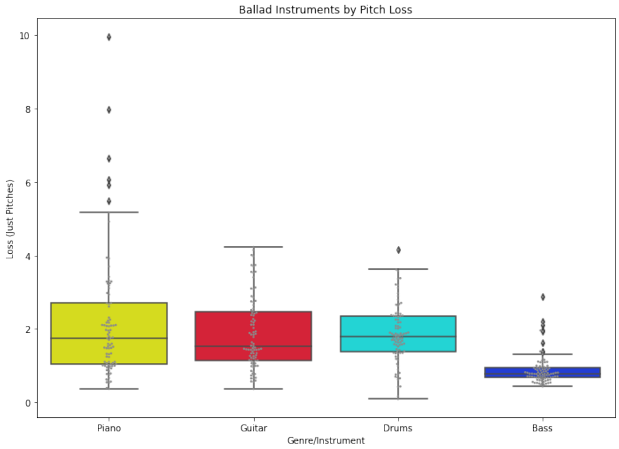


Figure 12. Ballad Genre Pitch Only Loss by Instrument

House had substantially worse loss in an inverse proportion to pop rock and ballad. In other words, while the loss was higher with the piano and guitar for pop rock and ballad, for house the loss was concentrated in the bass and drum instruments. Since these two instruments had the highest note incidence rate, we theorized that this contributed to the higher loss, however as you can see in figure 11 their distribution is quite different for each other with drums being much more tightly clustered. You can also see two main clusters in the bass distribution, as far as these loss functions relate to incidence rate these may represent the 'grooves' in relation to the constant BPM of the drums. However, there was also a substantial difference in loss for the velocity feature for house music in the rhythmic instruments: bass and drums (fig 14). There was a larger variance in velocity for the house drums than any other genre or instrument. Velocity loss was proportionally higher than pitch and duration loss for house drums, showing that this increased loss rate was not a result of only higher note incidence rate but also the diverse composition across velocity. This is logical when the distinctive sounds of the genre are considered because it often features layered percussion elements at different velocities used to create more complex sounds for the rhythm.

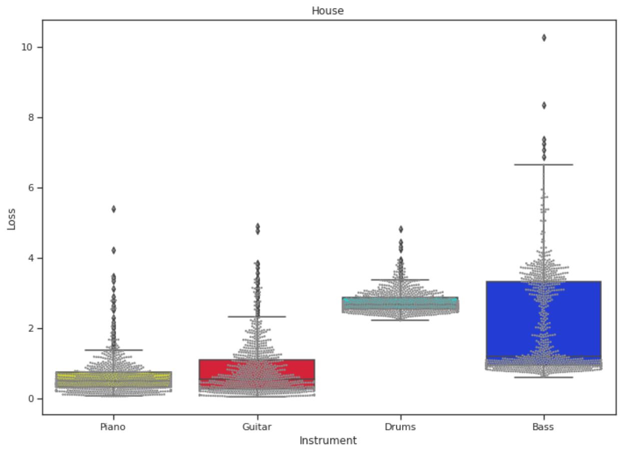


Figure 13. House Genre Loss by Instrument

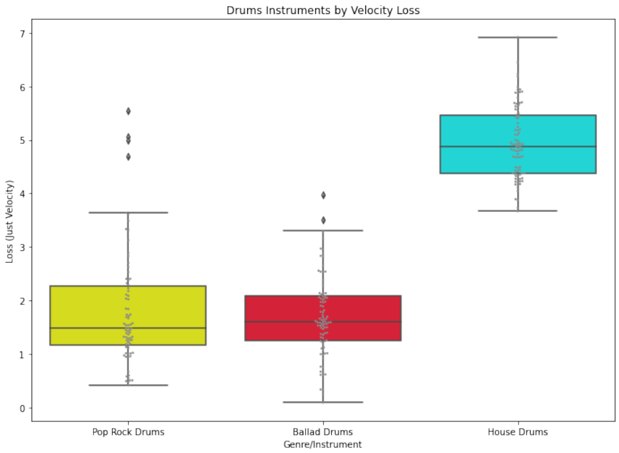


Figure 14. Drums Velocity Only Loss by Genre

6. DISCUSSION AND FUTURE WORK

We noticed several confounding factors in our work that made it difficult to achieve our results. First was the generally sparse nature of the data, there were many time steps without notes and many time steps with only one or two notes out of ten possible notes. We considered condensing the input to encode less possible notes at each time step, and less time steps in general, to make it more dense, but decided against this because our aim was to use the models to elucidate the musical data, rather than produce the most realistic possible output. In order to learn this sort of sparse data more effectively, ideally it would require a deeper model with more embedding layers and substantially more units in each layer, larger datasets, and more training iterations to further fit the data. Despite our constrictions on the computation resources used for this project, we were still able to return low loss with demonstrated feature learning as the loss broadly decreased throughout iterations and epochs.

Another source of error was differing sizes of datasets by genre, where pop rock and ballad each had 9914 and 7014 songs respectively, while house had only 3217. Though this is certainly an issue as it relates to the potential set of songs being less likely to describe the total variance of composition expressed in the house genre, we found that this size of data was still mostly sufficient for our smaller scale models. This was demonstrated by the largest decrease in loss occurring in the first epoch for each dataset regardless of size, so it was clear that despite the lower number of samples that the house models still were able to learn some of the features of the house music genre.

There are many possible possible paths forward to both improve this system's ability to generate coherent samples and explore the data more completely. One interesting area of inquiry might be to train models on all 3 genres at the same time to create a better baseline for the instruments and the effect of larger or smaller amounts of data. We might also look into analyzing more specific subsets of the data, such as the cluster of middling loss samples in the bass instrument for house music, to determine what structures are causing that change in loss value. We also would like to expand to model more genres in order to observe more links between broader style definitions. However due to our limited time and computational resources in combination with our complicated models, lots of potential ideas were out of reach for the scope of this project.

7. CONCLUSION

In summation, we were able to observe clusters of overlapping song genre tags, find a mapping of 128 instrument channels to 4 sub-groupings, and track significant dissimilarities in learnable features across both genres and instruments for our models. We categorized pop rock, ballads, and house music as the most promising genres due to their instrument distribution and that instruments tracks could be condensed into piano, guitar, drums, and bass. We also found that the melodic instruments of guitar/piano carried more complexity in the ballad and pop rock genres, while the more rhythmic instruments like drums and bass were more difficult to learn for the house genre. Inside of those sub-categories, the melodic instruments had higher loss in relation to pitch features, while the rhythmic instruments varied more strongly based on velocity. We successfully implemented a robust data pipeline from the million songs dataset, discerningly selecting songs/instrument clusters, and encoding types before feeding the data to 16 models of the same RNN architecture. We assert multiple unique findings coming from both the pipeline and models with a wide open future of research using modified versions of this underlying system.

8. REFERENCES

[1] T. Bertin-Mahieux, D. Ellis, B. Whitman, and P. Lamere. The million song dataset. 2011.

[2] D. Eck and J. Schmidhuber. Finding temporal structure in music: blues improvisation with lstm recurrent networks. In *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*, pages 747–756, 2002.

[3] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[4] N. Kotecha and P. Young. Generating music using an lstm network. 2017.

[5] C. Raffel. *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching*. PhD thesis, COLUMBIA UNIVERSITY, 2016.

[6] B. Sturm. A simple method to determine if a music information retrieval system is a “horse”. *IEEE TRANSACTIONS ON MULTIMEDIA*, 16(6), 2014.

[7] C. Pui Tang, K. Long Chui, Y. Kin Yu, Z. Zeng, and K. Hong Wong. Music genre classification using a hierarchical long short term memory (LSTM) model. In Xudong Jiang, Zhenxiang Chen, and Guojian Chen, editors, *Third International Workshop on Pattern Recognition*, volume 10828, pages 334 – 340. International Society for Optics and Photonics, SPIE, 2018.

[8] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10:293 – 302, 08 2002.

[9] Y. Zhou, W. Chu, S. Young, and X. Chen. Bandnet: A neural network-based, multi-instrument beatles-style midi music composition machine. In *ISMIR*, 2019.