#Rechorder: Anticipating Music Motifs in Real Time

Tommy Li Yash Savani Wilbur Yang

***Abstract*—In this paper, we explore the potential for simple statistical techniques in predicting music as a computer algorithm listens to music. We reduce a collection of MIDI files to a progression of note pitches and intensities on a single octave and run k-means clustering on the results. We then track statistical patterns in transitions between clusters to begin to build a model that attempts to predict the next cluster given all musical information before a point in time. We find that sequence matching works better than an SVM for predicting the next notes. This is likely because the SVM model is highly local to the given point in time. Future work on this topic should take account of instrument data and make a more naturally global prediction model.**

I. Introduction

A. Motivation

MUSIC is a fascinating art form. It is defined as the ordering of tones or sounds in succession, in combination, and in temporal relationships to produce a composition having unity and continuity.

If we break down this definition, we find that music is understood as a pattern or progression of sounds that sound harmonious when played together. Many of these patterns are ubiquitous throughout contemporary western music as chord progressions. Most accomplished musicians and music theoreticians can attest to common progressions that make up most of the music we hear. Furthermore, with some training and knowledge of the genre of music it is possible to predict, given the current state of the song, the next chord in the progression.

Our goal was to use machine learning to predict the next chord in the progression of music in a similar manner to how a human music aficionado would predict the next chord. By predicting the next chord, a computer can use that information to generate accompaniment tracks, create new music, and assist in composition, among other things, which can be of great benefit to musicians.

This project was done for both CS229 and CS221 at Stanford University.

B. Our Work

Early on in our research, we realized that there are several genres of music that do not rely on the western musical structure of chords. For instance, instead of chords, classical Indian music uses a movable seven-note scale called Ragas which differs from the scale system used in western music. We wanted our system to be comprehensive enough to be able to map all genres of music regardless of the scale system.

As a result, we chose to apply a form of unsupervised learning to break the music down into its defining motifs; in the case of western music, these motifs would be chords. We then wanted to predict transitions between those motifs. We decided to use k-means clustering to find the motifs of the music and then apply the SVM algorithm to predict the next motif.

C. Literature Review

Time series analysis and prediction is the target of significant research effort, although music prediction and generation is relatively less well-studied. Here are some interesting approaches to the task. Paiement *et al.* builds a feature mapping such that the Euclidean distance between the representations of chords approximately match empirical psychoacoustic differences between them, and then adds a graphical model on top of this representation [1]. Hu *et al.* models music with an unsupervised approach that attempts to find musical key profiles of various genres of music [2]. Lavrenko *et al.* examines music in terms of start times of notes and models the music using random fields [3]. In contrast with Lavrenko *et al.* we consider the instantaneous intensity of sounds. We do not attempt to make cases for separate genres and also do not build our model on known psychoacoustic differences.

II. Dataset

We used a collection of 60 pop songs scraped from MIDIWorld.com, originally performed by the artists Green Day, Taylor Swift, The Beatles and The Foo Fighters. The MIDI format essentially provides the start and stop times of various instruments and their volumes.

III. Models

We modeled the problem in two parts: the first is an unsupervised clustering problem, and the second is a supervised classification problem.

A. Unsupervised Learning

We extracted musical bars of equal width from all of the songs in our corpus of MIDI files. Each bar was represented as a 12 dimensional feature vector where each element represented the presence of the corresponding note. We clustered all of these bar vectors using the k-means algorithm. This process served to label each bar with a general notion of a motif. We then used these cluster numbers as the labels for our classification problem.

In our analysis of the model, we found that the optimal bar width is most likely to be four beats. This result follows from the assumption that we are training the algorithm on pop songs, which most likely have a $\frac{4}{4}$ time signature.

To select the best value of k, we ran k-means multiple times with different values of k and then graphed the average silhouette value for each value of k. We then selected the value of k that resulted in the highest average silhouette value. The silhouette value for a given data point is a representation of how well the clustering algorithm performed with respect to that point. We can find the silhouette value for any point by using

$$s(i) = \frac{b(i) - a(i)}{\max\{b(i), a(i)\}} \tag{1}$$

where $s(i)$ is the silhouette value for point i, $b(i)$ is the lowest average dissimilarity of i to any other cluster which i is not a member of, and $a(i)$ is the average dissimilarity of i with all other data within the same cluster. We picked k to be equal to 7 as will be elaborated on in the results.

B. Supervised Learning

The classification problem then becomes the following: given the labels of the previous five bars seen, what is the label of the next bar? We chose to perform the classification using a multiclass SVM, and we compared this method to an ad hoc sequence matching technique as well as a simple method which makes predictions by repeating the label of the current bar.

Alternatively, another model that we chose to employ was a Markov model which predicted the label (cluster) of the next bar, given the current bar. The algorithm estimated the probability of transitioning from one cluster to another empirically, given all of the transitions that it had seen so far. This model saw limited success and we did not follow up on it.

C. Features and Preprocessing

To perform clustering, we parsed the MIDI files into time segments of equal size, four beats per bar, to make the information more manageable. We discarded information regarding choice of instrument and the notes' octave in order to focus on pitch and chord. Because there are 12 notes in an octave, including sharps and flats, this normalization produced 12-dimensional vectors representing the intensity of each chord. These sequences of 12-dimensional vectors were later used throughoutout the project.

For the classification part of the project, we needed to classify the previous five bars of music. This data was a $k \times 5 = 35$ dimensional binary vector – each of the five groups of 7 dimensions represented the label of one of the previous five bars.

IV. Results

A. K-means

Figure 3 displays the results obtained from running k-means with different values of k and then plotting the average silhouette value. Although there does not seem to be a clear trend, there is a slight bump at $k = 7$ suggesting slightly better clustering. As a result we fixed $k = 7$ for the rest of the project. Figures 4, 5, and 6 illustrate the confusion matrices of the SVM, sequence-matching and repeating approach respectively.

Table I lists the results obtained from 10-fold cross validation.

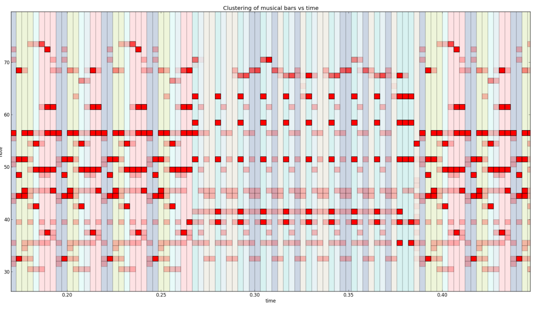

Fig. 1. A visualization of the MIDI file cut into clustered bars. Each bar, a sequence of four consecutive beats, is given a color that represents its cluster. The red rectangles are the notes present during each bar, and their opacity represents the duration for which they are being played within the bar.

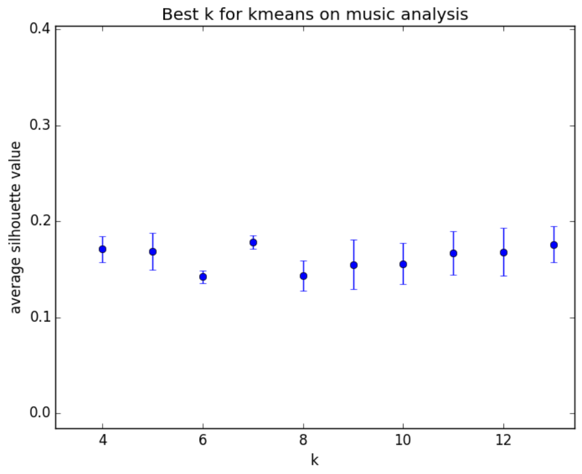

Fig. 3. Comparison of silhouette value for various choices of k when clustering all data. There is a small jump at $k = 7$.

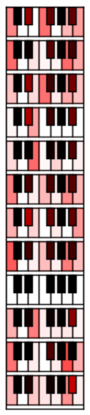

Fig. 2. A piano chord representation of 12 discovered cluster centroids. Each chord can be interpreted as a generalized music motif in the corpus of MIDI files that we trained the algorithm on.

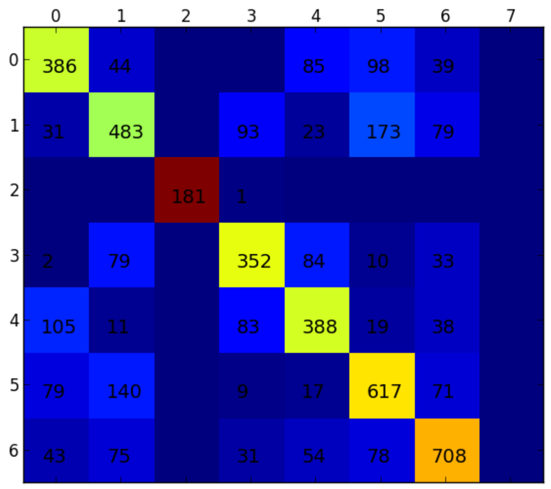

Fig. 4. Confusion matrix for the SVM.

TABLE I
RESULTS OF 10-FOLD CROSS-VALIDATION

Prediction approach	Training accuracy	Testing accuracy
Support Vector Machine	0.660	0.640
Sequence-matching	N/A	0.654
Predict stationary	N/A	0.615

V. DISCUSSION

Our experiments with our models show that Support Vector Machines fed the previous 5 clusters do not seem to do significantly better than a baseline model in which after any given time we predict that the next cluster will be the same as the current one. Sequence matching, in other words, seeking the longest sequence of previously-heard clusters that matches the current cluster, seems to have a slight predictive edge over always predicting the same cluster as the previous one.

There are various factors that may be limiting the quality of the predictions. One factor is that clusters may not land in zones that have emotional meaning. For instance, given that we were testing on pop music, we could have employed some simple assumptions about what chords had significance in pop music to select "good" clusters in advance. However, since the intention for our model was to be culturally agnostic we let the clusters arise in a to-

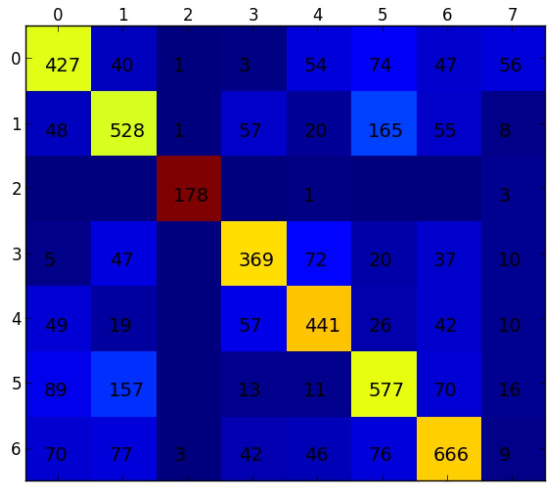

Fig. 5. Confusion matrix for the sequence-matching algorithm.

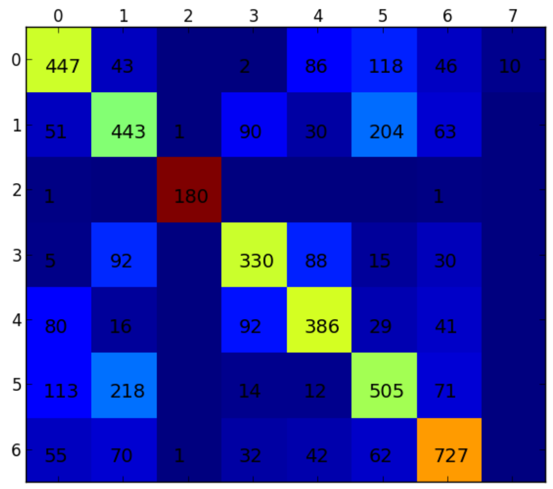

Fig. 6. Confusion matrix for the baseline repetition approach.

tally unsupervised manner. Perhaps the assumption that there exist universal clusters is too reductive. It may that be the most optimal choice is a somewhat more complex model in which we find important clusters in various well-known genres of music and identify the genre before we predict the cluster.

It may also be possible that our feature mapping loses too much information of human significance. For instance, our model has no access to information about instrument use, and cannot distinguish between, say, a flute solo and a bass backing. Incorporating instrument use (i.e. timbre) may yield more interesting results.

The relative success of sequence-matching approaches also suggests that a greater integration of search-like algorithmic approaches with our conventional statistics-based approach may be useful. In particular, examining our decomposition of a song as seen in Figure 1, it is visually obvious that there are highly repetitive sequences of notes, rather than repeated sequences of clusters. An approach that decomposes a MIDI file into repeated motifs in a pre-processing step may have greater success and more explicit predictive power.

VI. Conclusion

In our explorations, the most effective method for predicting the next cluster that music will lie in is to use sequence matching, which outperforms predicting no change as well as the multiclass SVM. Our sequence-matching algorithm outperformed more naive predictions by a small margin, suggesting that there is more to be desired in feature extraction. It remains to be seen what the clusters found actually mean. The framework for analysis of MIDI files that we employ here is quite customizable. Moreover, it seems to be a ripe area to employ prediction algorithms: in contrast with full mp3 files, analysis of MIDI files gives us direct access to the discrete notes of the songs and allows for relatively easy integration of more complex pre-processing procedures.

VII. Future

We have found an effective technique to try and predict motifs and motif progression. However there is still a lot of work that can be done to try and expand on what we have already found. Firstly we make an assumption that the tempo of the song is constant and that the key signature of the song is always $\frac{4}{4}$. However, there are still examples of songs that do not follow the classic common time signature, and for several songs, the tempo of the song may change as the song is played. We could try to use additional supervised learning techniques to model these assumptions to account for the different types of music. Another area we can try to improve in is the off beat notes that often appear before the beginning of the first complete bar of the song. These off beats occur in many songs as they build up to the official beginning of the song. Again we could use a supervised learning algorithm to try and predict the first beat of the song and then use that to generate our bars. Also, as discussed in the Discussion section, we realized that, due to the high percentage of recurring motifs, the learning algorithms learnt to generally select the same motif for the next progression. To account for this, we could use regularization to decrease the effect of the recurring motifs on the trainer.

REFERENCES

[1] J. Paiement, D. Eck, S. Bengio. A Probabilistic Model for Chord Progressions. http://bengio.abracadoudou.com/cv/publications/pdf/paiement_2005_ismir.pdf

[2] D. J. Hu, K. Saul. A Probabilistic Topic Model for Unsupervised Learning of Musical Key-Profiles. http://cseweb.ucsd.edu/~saul/papers/ismir09_lda.pdf

[3] V. Lavrenko and J. Pickens. Polyphonic music modeling with random fields. In Proceedings of ACM Multimedia, Berkeley, CA, November 2-8 2003.

[4] Various. Collection of MIDI files based on well-known pop songs. www.MIDIWorld.com

[5] Pedregosa *et al.*, Scikit-learn: Machine Learning in Python, in JMLR 12, pp. 2825-2830, 2011.