Generating Groove: Predicting Jazz Harmonization

Nicholas Bien (nbien@stanford.edu)
Lincoln Valdez (lincolnv@stanford.edu)

December 15, 2017

1 Background

We aim to generate an appropriate jazz chord progression for a given melody. As musical minds, we wanted to see if it is possible to create a model that can make any melody sound jazzy. Such a system could be used to generate alternative harmonizations for existing pieces or as an aid to help composers choose a chord progression for their melodies.

1.1 Related Work

Prior research has generated chord harmonizations for classical and pop music with high accuracy[1,2]. However, jazz music is much less formulaic than these genres. A study on jazz chord prediction has study achieved ≈50% accuracy in prediction, albeit with heavily generalized chord types and a very small, homogeneous data set[3]. We were unable to find a study that attempts to tackle the problem of predicting jazz chords from a melody. Our approach for incorporating both melody notes and chord transitions takes inspiration from Cunha et al.[4] and Paiement et al.[5]

2 Experiments

2.1 Dataset

Our dataset was a set of 149 Jazz Standards encoded in Lilypond musical transcription format[6]. Each song in the dataset consists of a melody line with accompanying chord symbols. We transposed each song to the key of C major/A minor. For each chord symbol, we count the number of times each of the 12 notes in the chromatic scale occurs to obtain our features. Since chord changes can occur at any time, we chose to use the same chord positions as our dataset. Thus, notes were grouped based on which chord they appear under in the music.

2.2 Chord Clustering

There were 246 chord types in our dataset. To facilitate classification, we explored methods for grouping chords into similar clusters.

We first tried using K-means clustering with the normalized note counts as features and seeing which chords centered around the initialized centroids. K-means in this case took the normalized vector of note counts observed per each chord class and identified clusters by minimizing chord-to-chord distances in the 12-dimensional note space.

Our second method was Brown clustering, which is commonly applied to natural language to cluster words based on their use in context. The algorithm clusters tokens based on bigrams: if two tokens frequently appear before or after some other token, they are likely to be clustered together. In our case, we wanted to discover which chords had similar functions: certain basic chord transitions, such as ii-V-I, are extremely commonly in jazz, but the exact chords used in such progressions varies.

Combining the results of the two algorithms, we observed that chords where notes were stacked on top of the seventh were generally clustered together, and that seventh chords had similar functions to their respective triads and sixths. Ultimately, due to noise in the clustering data, we chose to hand-pick our chord clusters. For each of the twelve possible root notes, we grouped chords into varieties of sevenths: major, minor, dominant, diminished, half-diminished, and augmented. Our clustering efforts brought the number of chord classes down from 246 to 85.

2.3 Chord Prediction

Our primary goal was to predict which chord best accompanies each group of notes in a song. We began by converted all chords in the dataset to their most similar seventh chord based on our clustering heuristic. For this task we explored two basic supervised learning models, then combined the better of these two models with a sequence prediction model.

2.3.1 Naive Bayes

For our first chord prediction algorithm, we implemented a Naive Bayes classifier with add-one Laplace smoothing. In analogy to document classification, we treat notes as words and chord types as document classes. The output is for each note group is:

$$\arg\max_y \prod_i p(y|x_i),$$

where y is the output chord and each x_i is a note in the note group. The classifier learns characteristic note distributions for each chord class from the training set, then uses these distributions to predict chords for the test set.

2.3.2 Support Vector Classifier

Our second method was a multi-class support vector classifier. Input vectors were 12-dimensional note counts. We obtained the best results with a radial bias function kernel, which for two note vectors x and x' is:

$$K(x, x') = \exp \frac{||x - x'||^2}{2\sigma^2}.$$

The classifier fits a series of hyperplanes to the data which aim to maximize the distance between vectors with different class labels.

2.3.3 SVC with Hidden Markov Model

Lastly, we tried combining a hidden Markov model with our SVC to capture relationships between chords[5,6]. For this task, we augmented the dataset with START and END tokens. The Markov transition probabilities were calculated from bigrams in the training set. The probability of transitioning from a chord y_1 to chord y_2 is:

$$p(y_2|y_1) = \frac{p(y_1, y_2)}{p(y_1)},$$

where $p(y_1, y_2)$ is fraction of y_1, y_2 bigrams in the training set and $p(y_1)$ is the fraction of all bigrams beginning with y_1 in the training set.

To incorporate the note features with the Hidden Markov Model, we used SVC outputs as noisy estimators of the ground truth chord to obtain our emission probabilities. We run the SVC on the training data to obtain its predictions for each chord. Then, we calculate the probability that the ground truth chord is y given that we observe the SVC prediction $\hat{y}$ using Bayes' rule:

$$p(y|\hat{y}) = \frac{p(\hat{y}|y)p(\hat{y})}{p(y)},$$

where $p(\hat{y}|y)$ is the fraction of times the SVC predicted $\hat{y}$ when the ground truth was y, $p(\hat{y})$ is the total fraction of times the SVC predicted $\hat{y}$, and $p(y)$ is the total fraction of times chord y appeared in the training data. We then found the maximum likelihood sequence of hidden states (chords classes) given the observed states (SVC predictions) in the test data using the Viterbi algorithm.

3 Results

For testing, we used 80/20 cross validation (split by song). Depending on the run, this amounted to ≈6200 chords in the training set and ≈1500 chords in the test set. Our baseline is to always predict the most common chord (Cmaj7). The test accuracy was as follows:

Baseline	17.6%
Naive Bayes	26.2%
SVC	28.3%
SVC+HMM	27.4%

3.1 Analysis

All algorithms outperformed the baseline. Our low accuracy is explained by a variety of factors including: our high number of chord classes (even after clustering), sparsity of features (many note groups contain only a single note), and quite frankly the nature of jazz music (in a genre rules are made to be broken, the melody sometimes diverges completely from its chord progression).

Despite this, we were generally pleased with the aural quality of the harmonizations produced. In particular, incorporating a Hidden Markov Model with the SVM lowered overall accuracy slightly but produced more canonical jazz progressions, even for non-jazz music. Figures 1 and 2 show two harmonizations of the ubiquitous "Happy Birthday". These simple examples reveal the most important difference between our SVC model and our SVC+HMM model.

The first harmonization, using SVC, makes predictions based solely off notes. Measures containing mostly notes in the C triad are harmonized with Cmaj7 (which is in the same cluster as C), while measures containing mostly notes in the G7 chord are harmonized with G7. The SVC output matches the most common harmonization for this melody.

The second harmonization, using SVC+HMM, makes predictions based off notes as well as chord transitions. Thus, it doesn't match the notes as well, but predicts two ii-V-I sequences, the most common progression in jazz. This harmonization technically isn't as correct as the SVC's harmonization, but it does add a jazzy feel.

One key thing to keep in mind is that music, especially jazz music, is up to interpretation; what could sound beautiful to one ear could sound displeasing to another. As such, test accuracy is not always the best measure of success. Another metric measured was the

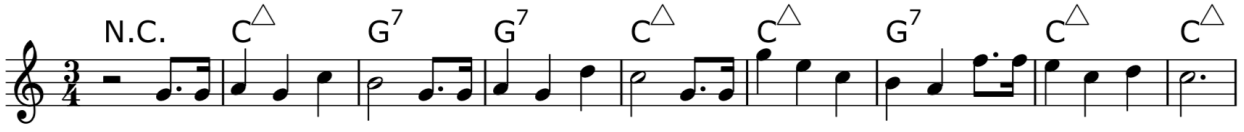

Figure 1: Harmonization of "Happy Birthday" with SVC

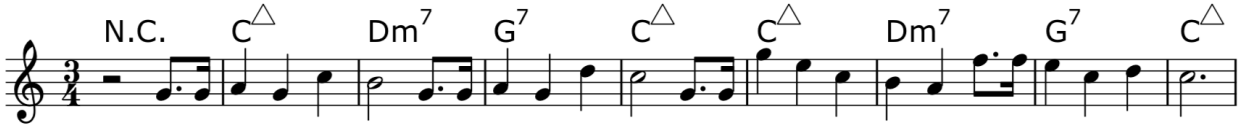

Figure 2: Harmonization of "Happy Birthday" with SVC+HMM

distribution of chords predicted, to see if our model was truly learning anything. For each model and the dataset as a whole, the percentages predicted were as follows:

	Overall	Naive Bayes	SVC	SVC+HMM
Cmaj7 (I)	17.6%	51.2%	56.8%	40.7%
G7 (V)	11.4%	1.0%	0.7%	21.1%
Dm7 (ii)	11.2%	20.9%	26.5%	12.6%
Am7 (vi)	8.2%	5.1%	5.1%	9.8%
E7 (V of Am7)	4.5%	4.3%	2.4%	3.5%

Our Naive Bayes and SVC models output accurate distributions for four of the top five most commonly seen chords in jazz music. However, these models severely under-predict the second most common chord, G7. We suspect that this is because G7, as the dominant seventh chord in C major, is ripe for extensions like flat-9ths and sharp-11ths. These alterations are all in our G7 cluster, but each one may best accompany much different sets of notes. Our SVM+HMM corrects this issue by taking into account chord transitions. Since G7 commonly precedes Cmaj7, it is predicted much more often by our SVM+HMM model.

In general, quantitative metrics are not the best measure for music. Success is best measured by whether the harmonizations are aurally pleasing. Each of our models succeeded in this regard; chords usually matched well with the melody and, with the SVC+HMM model in particular, transitions were smooth. All three algorithms, when given melodies from other genres of music (such as pop or classical) were able to generate a pleasing result.

4 Future Work

The problem of jazz harmonization turned out to be even more complex than we originally anticipated. In comparison to other genres, algorithms for understanding and predicting jazz music are limited. We likely could have achieved much higher accuracy with the same approach on classical music, since jazz music is far too expansive given the limited data set we had access to. In this vein, gathering more data would almost certainly improve our algorithm. We also see potential in the ability of neural networks to learn more intricate note-to-chord relationships, as well as longer-term chord-to-chord relationships that our models couldn't capture. There is still much work to be done before computers are able to rival the best jazz composers.

5 References

1. Harbuz. Modelling Classical Music with Machine Learning. 2016.

2. Simon, Morris, and Basu. MySong: Automatic Accompaniment Generation for Vocal Melodies. CHI Proceedings. 2008.

3. Thom and Dannenberg. Predicting Chords in Jazz. ICMC Proceedings. 1995.

4. Cunha, Sidney, and Ramalho. An Intelligent Hybrid Model for Chord Prediction. Organized Sound 4(2), pp. 115-119. 1999.

5. Paiement, Eck, and Bengio. Probabilistic Melodic Harmonization. LNAI 4013, pp. 218-229. 2006.

6. https://github.com/veltzer/openbook

6 Contributions

The first author wrote the parser and implemented Brown clustering and the SVC+HMM model. The second author implemented K-means clustering and the Naive Bayes and SVC models. Both authors contributed equally to this paper.

Code: https://github.com/nicholasbien/dj-jazzy-jeff