Robust modeling of musical chord sequences using probabilistic N-grams

Ricardo Scholz, Emmanuel Vincent, Frédéric Bimbot

ROBUST MODELING OF MUSICAL CHORD SEQUENCES USING PROBABILISTIC N-GRAMS

Ricardo Scholz, Emmanuel Vincent, Frédéric Bimbot

METISS Project Team – IRISA / INRIA / CNRS
Campus de Beaulieu – 35042 Rennes Cedex – France
ricardoscholz@gmail.com, evincent@irisa.fr, bimbot@irisa.fr

ABSTRACT

The modeling of music as a language is a core issue for a wide range of applications such as polyphonic music retrieval, automatic style identification, audio to symbolic music transcription and computer-assisted composition. In this paper, we focus on the modeling of chord sequences by probabilistic N-grams. Previous studies using these models have achieved limited success, due to overfitting and to the use of a single chord labeling scheme. We investigate these issues using model smoothing and selection techniques initially designed for spoken language modeling. This approach is evaluated over a set of songs by The Beatles, considering several chord labeling schemes. Initial results show that the accuracy of N-grams is increased but that additional improvements may still be achieved in the future using more advanced, possibly music-specific, smoothing techniques.

Index Terms— Music, probabilistic modeling, N-grams, model smoothing, model selection.

1. INTRODUCTION

Music has several dimensions: melody, harmony, rhythm... The harmony dimension consists of a sequence of chords, represented by symbols from a finite dictionary. The modeling of chord sequences is useful for many tasks, such as polyphonic symbolic music retrieval [1, 2], composer characterization [3], chord transcription from audio [4] and computer-assisted composition or harmonization [5].

Considering that chord sequences exhibit strong short-term dependencies, this problem has often been studied by modeling sub-sequences of N successive chords termed N-grams. Existing approaches include deterministic N-grams [1] and probabilistic 2-grams, i.e., Hidden Markov Models (HMMs) [4]. Deterministic models offer a computationally efficient approach to retrieval applications but cannot model the likelihood of any chord sequence, as needed for certain other applications. HMMs address this issue, but their low model order $N=2$ does not match the actual complexity of music. In this article, we investigate the modeling of chord sequences via probabilistic N-gram models with $N \geq 2$. Such models have been used for other types of data, such as melody [5] and spoken language [6]. The first attempts to model chord sequences via this approach [2, 3, 7] achieved limited success, due to fixed model inputs and parameters and to overfitting issues.

In the following, we aim to address these issues by applying model smoothing and selection techniques and by considering different chord dictionaries. Training strategies for probabilistic N-grams are discussed in Section 2. The chosen chord labeling schemes and preprocessing steps are then presented in Section 3. Initial results are described in Section 4. Finally, conclusions are given in Section 5.

2. PROBABILISTIC N-GRAM MODELING

2.1. Model Definition

Consider a song S which consists of a sequence of $|S|$ chords labeled $C_1, C_2, \ldots, C_{|S|}$. The likelihood of S is defined as

$$P(S) = P(C_1, \ldots, C_{|S|}) \quad (1)$$

We assume that each chord C_i depends only on the $N-1$ previous chords $C_{i-N+1}, \ldots, C_{i-1}$, called the truncated history of C_i. The likelihood of S is therefore modeled as

$$P(S) = P(C_1, \ldots, C_{N-1}) \cdot \prod_{i=N}^{|S|} P(C_i \mid C_{i-N+1}, \ldots, C_{i-1}) \quad (2)$$

where the initial term is the probability of a particular chord sequence of length $N-1$ to occur at the beginning of a song and the general term is the conditional probability of a particular chord to occur at any other time given each possible truncated history.

In order to compare the likelihood of chord sequences of different lengths, the normalized negative log-likelihood, also called *perplexity* [6], is often used instead:

$$\bar{P}(S) = -\frac{\log_2 P(S)}{|S|} \quad (3)$$

This quantity is expressed in bits per symbol. The perplexity of a set of songs H is then computed as:

$$\bar{P}(H) = -\frac{\sum_{S \in H} \log_2 P(S)}{\sum_{S \in H} |S|} \quad (4)$$

2.2. Training Issues

Maximum likelihood training aims to estimate the set of model probabilities such that $\bar{P}(H)$ is minimum over some training set H (distinct from the test set). This is achieved [6] by setting the transition probabilities to

$$P_{ML}(C_i|C_{i-N+1},\dots,C_{i-1}) = \frac{\mathrm{c}(C_{i-N+1},\dots,C_i)}{\sum_{C_k}\mathrm{c}(C_{k-N+1},\dots,C_k)} \quad (5)$$

where $c(C_{i-N+1},\dots,C_i)$ is the number of occurrences of the N-gram $C_{i-N+1},\dots,C_i$ in the training set. Initial probabilities are obtained similarly [6].

When the training set is small compared to the number of possible N-grams, the test set is likely to contain sequences of symbols that were not observed over the training set, resulting in infinite perplexity. More generally, the existence of sequences with few observations in the training set leads to *overfitting*, i.e., larger perplexity values over the test set. In order to achieve better generalization, smoothing must be applied to the learnt probabilities, so that they may better predict non-observed set of data.

This issue was acknowledged in [2] and addressed by learning a Universal Background Model (UBM) from a superset of the training data and interpolating it with the ML model. This approach is impractical for some applications where additional training data may not be available. Also, the UBM itself may suffer from overfitting.

We have used two smoothing techniques originally designed for spoken language modeling: additive smoothing and Jelinek-Mercer (JM) smoothing [6]. Additive smoothing consists of adding a positive number δ_N to the count of each possible sequence of symbols before performing a normalization of the counts. This is equivalent to assuming that each N-gram occurs δ_N times more than it actually does in the training set:

$$P_{add}(C_i|C_{i-N+1},\dots,C_{i-1}) = \frac{\delta_N + \mathrm{c}(C_{i-N+1},\dots,C_i)}{\delta_N\cdot|D| + \sum_{C_k}\mathrm{c}(C_{k-N+1},\dots,C_k)} \quad (6)$$

where D is the considered dictionary of chords symbols and $|D|$ the number of symbols in D.

JM smoothing is more sophisticated as it interpolates higher-order N-gram models from lower-order N-gram models. Indeed, when there is insufficient data to estimate a probability in the higher-order model, the lower-order model can often provide useful information. The interpolation is done recursively as follows:

$$P_{JM}(C_i|C_{i-N+1},\dots,C_{i-1}) = \lambda_{N-1}\cdot P_{ML}(C_i|C_{i-N+1},\dots,C_{i-1}) + (1-\lambda_{N-1})\cdot P_{JM}(C_i|C_{i-N+2},\dots,C_{i-1}) \quad (7)$$

where λ_{N-1} is a real-valued number between 0 and 1. The recursion starts with the smoothed 0th-order model, which is assumed to be the uniform distribution.

The best model can be estimated by testing all possible combinations of smoothing techniques, parameter values δ_N and $\lambda_0,\dots,\lambda_{N-1}$ and model orders N and selecting the one leading to minimum perplexity over the test set.

3. CHORD LABELING SCHEMES

There are several chord labeling schemes in music, and the choice of one instead of another depends mainly on the music style studied [8]. Perplexity values are influenced by two main factors: the number of different symbols in the dataset and their meaningfulness. The labeling scheme has strong influence on both of these characteristics. Hence we consider six different schemes in the following.

3.1. Tonality-dependent schemes

Harte [8] has proposed a grammar for chord labeling, which he claims to be simple and intuitive for musically trained individuals to write and understand. Chord symbols are defined by a list of properties: root, type and inversion. The root is the name of the note upon which the chord is built. All possible note names are accepted, including enharmonic names, e.g. *D#* and *Eb*. The type property lists the pitch intervals making up the chord relative to the root. It typically consists of one of 17 common *shorthand types*, plus possible dissonant intervals. If no dissonances are made explicit, the chord contains only the notes defined by its shorthand type. The inversion property is the pitch interval between the root and the bass note. If no bass information is provided, the root is assumed to be the bass note, as usual in musical notation. For example, the symbol *G:maj7(#9)/3* denotes a *G* major seventh chord with augmented ninth, i.e. the following set of notes: {*G, B, D, F#, A#*}, with *B* as the bass. The symbol *G:min* denotes a *G* minor triad, i.e., {*G, Bb, D*}, with *G* as the bass. In the following, we consider Harte's original grammar, except that we consider enharmonic roots to be equivalent, resulting in a set of 12 possible roots. Although the number of possible chord symbols is on the order of several million, 392 symbols were observed in the data of Section 4.

Most of the current approaches for chord-related tasks assume a set of 24 chords instead, consisting of major and minor chords from 12 possible non-enharmonic roots [2, 4]. Usually, as the minor/major labeling is oversimplified for practical applications, the chords which are not exactly pure minor or major are transformed. Dissonances and bass information are discarded and the roots are respelled so as to remove enharmonics. Augmented or diminished/half-diminished chords are transformed into major and minor chords respectively.

In addition to the above two previously studied chord labeling schemes, we consider an intermediate scheme. It consists of representing a chord by its root and type, chosen among 12 non-enharmonic roots and the 17 shorthand types defined by Harte [8], but discarding dissonances and bass information which are musically less important. This leads to 204 different symbols. This scheme can model musical

information in a more realistic way than just considering major and minor chords, with a much smaller number of symbols than Harte's original labeling scheme.

3.2. Tonality-independent schemes

For each of the above three labeling schemes, we also propose a transformation function, which takes the labels and the song initial tonality as inputs and derives tonality-independent labels. The new labels consist of the exact same properties except for the root, which is replaced by one of 12 tonality-independent roots, representing its relation to the initial tonality. For example, the label *Emin7(b9)* in a song with initial tonality *D* major and the label *Amin7(b9)* in a song with initial tonality *G* major will both be transformed into *IImin7(b9)*. These labels provide better modeling of equivalent chord sequences and tonality modulations in different songs, even if they are in different initial tonalities. This contributes to a better generalization ability of the model, since a given chord sequence observed in a particular tonality contributes to the estimation of equivalent chord sequences transposed in different tonalities. The number of possible tonality-independent chord symbols derived from Harte's scheme is again on the order of several million, among which 349 were observed in the data of Section 4.

4. EXPERIMENTS

4.1. Dataset

The chosen dataset involves 14194 chord occurrences stemming from the 180 songs making up the 13 studio albums of The Beatles, labeled by Harte using the grammar in [8]. This dataset is by far the largest available today and is well suited to the study of language models since it covers a single music style. In order to allow the derivation of tonality-independent chord labels from this dataset, we appended to the original annotations the initial tonality of each song. The above tonality-dependent labeling schemes and their tonality-independent variants resulted in a total of six sequences of chord labels for each song.

4.2. Test and validation procedure

N-gram models were trained up to order $N=5$ for the two minor/major labeling schemes and up to $N=3$ for the four other schemes. Due to memory issues, we could not perform tests with higher-order models. The smoothing parameters δ_N were sampled from 22 logarithmically-spaced values between 0.01 and 5 and the parameters $\lambda_0,\ldots, \lambda_{N-1}$ from 20 values between 0.01 and 0.995.

The quality of the models was assessed by 13-fold cross-validation. In order to avoid the potential "album effect" [9] due to variation of the musical style of the Beatles over the years, each of the 13 albums was successively considered as the test set, with the remaining 12 albums composing the training set.

For each fold, four perplexity values were computed, considering:

(1) all N-grams in the training set,
(2) all N-grams in the test set,
(3) only the N-grams in the test set that were observed in the training set,
(4) only the N-grams in the test set that were not observed in the training set.

For each labeling scheme, the best model was selected as the one with lowest perplexity over the test set (2), averaged all folds. Ideally, this model should fit the test data as well as the training data, resulting in similar perplexity values (1) and (2). Hence the closer these two values, the more robust the model is to generalization. The additional perplexity values enable more detailed benchmarking, distinguishing between the generalization of the model to observed data and to non-observed data.

4.3. Results

Using JM smoothing, the best results were obtained for the largest tested order N, that is $N=5$ for the minor/major labeling schemes and $N=3$ for other schemes. By contrast, the best results using additive smoothing were always achieved with $N=2$, that is the order considered in most previous studies [4]. Higher-order models are consistent with music theory, which suggests that the dependency between chords in a sequence is often greater than only the previous chord. This shows that advanced smoothing techniques play a key role in attempting to reach the actual complexity of music.

Figure 1 depicts the four perplexity values computed for the tonality-independent minor/major labeling scheme using either no smoothing or one of the two smoothing techniques. JM smoothing reduces the perplexity over the test set by 0.5 bit/symbol compared to additive smoothing. This perplexity reduction can be seen to occur both over observed and non-observed N-grams, denoting good generalization even to significantly different data. Nevertheless, the perplexity over non-observed data remains 0.6 bit/symbol larger than over observed data, suggesting that further improvements might be achieved using more advanced, possibly music-specific, smoothing techniques.

The perplexity values obtained over the test set for all labeling schemes are listed in Table 1. Tonality-independent labeling reduces perplexity by 0.7 bit/symbol on average compared to tonality-dependent labeling. This is much larger than the quantity of information added by annotating the initial tonality of each song, that is $\log_2 12$ divided by the number of chords of that song or less than 0.05 bit/symbol. Thus tonality-independent labeling significantly improves the generalization capabilities of the model, as expected from music theory.

We also observe that the perplexity for shorthand type labels is about 1.0 bit/symbol larger than for minor/major

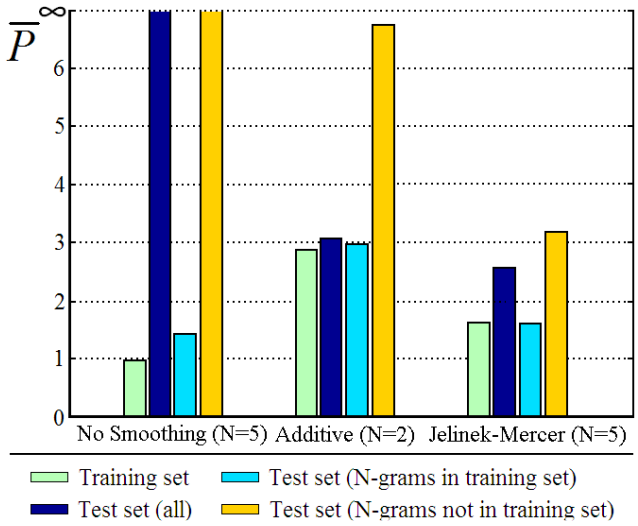

Figure 1 – Perplexities for the tonality-independent minor/major labeling scheme, using the best model for each smoothing technique.

Labeling scheme	**Tonality-dependent**	**Tonality-independent**
Minor/major	3.1925	2.5463
Shorthand types	4.2049	3.4908
Harte's with enharmonic equivalence	5.1450	4.3028

Table 1 – Perplexities over the test set, using Jelinek-Mercer smoothing and the best model for each labeling scheme.

labels. This difference is smaller than that expected from the increase in the number of symbols, that is $\log_2(204/24) \approx 3.1$ bits/symbol. However the difference in perplexity between shorthand type labels and Harte's labels is on the order of 0.9 bit/symbol, which is consistent with the doubling of the number of symbols. Therefore the model was able to predict shorthand types to a larger extent than their underlying minor/major triads, but not dissonances and inversions.

5. CONCLUSIONS

We studied the modeling of musical chord sequences via probabilistic N-grams, focusing on the improvement of the robustness of the models to different data. We showed that tonality-independent chord labeling and advanced model smoothing techniques are crucial to achieve good generalization capabilities and reduce perplexity compared to standard HMMs. We also found that the modeling of more complex chord types than the usual minor/major chords is feasible. This opens many research perspectives, whereby new smoothing techniques, possibly based on musicological expertise, could further improve the quality of the models up to rendering the actual complexity of music as a language. Implementation and memory issues are also worth considering so as to increase the model order. This research could lead to significant advances in several fields, including music information retrieval, audio to symbolic music transcription and computer-assisted composition.

6. REFERENCES

[1] S. Doraisamy and S. Rüger, "Robust polyphonic music retrieval with N-grams", Journal of Intelligent Information Systems, 21(1), pp. 53-70, 2003.

[2] E. Unal, P.G. Georgiou, S.S. Narayanan and E. Chew, "Statistical modeling and retrieval of polyphonic music", Proceedings of the IEEE Workshop on Multimedia Signal Processing (MMSP), pp. 405-409, 2007.

[3] M. Mauch, S. Dixon, C. Harte, M. Casey and B. Fields, "Discovering chord idioms through Beatles and Real Book songs", Proceedings of the International Conference on Music Information Retrieval (ISMIR), pp 255-258, 2007.

[4] H. Papadopoulos and G. Peeters, "Large-scale study of chord estimation algorithms based on chroma representation and HMM", Proceedings of the International Workshop on Content-Based Multimedia Indexing (CBMI), pp. 53-60, 2007.

[5] P.P. Cruz-Alcazar and E. Vidal-Ruiz, "Modeling musical style using grammatical inference techniques: a tool for classifying and generating melodies", Proceedings of the International Conference on Web Delivering of Music (WEDELMUSIC'03), pp. 77, 2003.

[6] S.F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling", Technical Report TR-10-98, Computer Science Group, Harvard University, 1998.

[7] J. Pickens and C.S. Iliopoulos, "Markov random fields and maximum entropy modeling for music information retrieval", In Proceedings of the International Conference on Music Information Retrieval (ISMIR), pp. 207-214, 2005.

[8] C. Harte, M. Sandler, S. Abdallah and E. Gómez, "Symbolic representation of musical chords: a proposed syntax for text annotations", Proceedings of the International Conferences on Music Information Retrieval (ISMIR), pp. 66-71, 2005.

[9] Y.E. Kim, D.S. Williamson and S. Pilli. "Towards quantifying the "album effect" in artist identification". Proceedings of the International Conference on Music Information Retrieval (ISMIR), Canada, pp 393-394, 2006.