A Data-Driven Approach to Chord Similarity and Chord Mutability

Dimitrios Bountouridis
Department of Information
and Computing Sciences
Utrecht University

Hendrik Vincent Koops
Department of Information
and Computing Sciences
Utrecht University

Frans Wiering
Department of Information
and Computing Sciences
Utrecht University

Remco C. Veltkamp
Department of Information
and Computing Sciences
Utrecht University

***Abstract*—Assessing the relationship between chord sequences is an important ongoing research topic in the fields of music cognition and music information retrieval. Heuristic and cognitive models of chord similarity have been investigated but none has aimed to capture the collective perception of chord similarity from a large dataset of user-generated content. Devising a large-scale experiment to gather sufficient data from human subjects has always been a major stumbling block. We present a novel chord similarity model based on a large amount of crowd-sourced transcriptions from a popular automatic chord estimation service. We show that our model outperforms heuristic-based models in a song identification task. Secondly, a model of chord mutations based on a large amount of crowd-sourced cover songs transcriptions is introduced. From crowd-sourced data, we create substitution matrices that capture the perceived similarity and mutability between chords. These results show that modelling the collective perception can not only substitute alternative, sophisticated models but also further enhance performance in various music information retrieval tasks.**

I. Introduction

Since the increase in popularity of blogging and social media sharing sites in the early 2000s, people started to contribute content to on-line crowd-source platforms with ever increasing amounts. With the vast amount of user generated content these platforms generate, researchers have proposed to harvest their inherent *crowd knowledge*. In this paper, we propose to capture crowd knowledge in a novel model of chord similarity that is based on a large collection of user generated chord sequences.

Chords are fundamental entities of the western tonal system to which a compact representation can be assigned. Consequently, the field of Music Information Retrieval (MIR) has been very prolific in terms of chord related research. The fact that chords have been successfully used as mid-level features to aid various tasks such as cover song detection, genre classification and others, has placed them, alongside automatic chord estimation (ACE), in the center of MIR research.

On-line chord transcriptions (either automatically or manually created) have been dominating the interest of music professionals and amateurs. Their active contribution to music websites has resulted in vast amounts of on-line databases with chord sequences. Probably the most popular on-line databases of manual chord annotations currently is *UltimateGuitar.com*. It offers to its users the ability to upload their own chord transcriptions, typically accompanied by lyrics. One of the most popular services providing automated, audio-derived chord transcriptions is *Chordify*, which is used by 1.5 million musicians every month. *Chordify* uses the HarmTrace [1] model to extract the chords from the audio. Most interestingly though, users are allowed to edit chords that they consider incorrect, effectively allowing them to create a personalized version of the ACE output.

Assessing the relatedness between chords sequences has been typically performed using pairwise alignment. It has been shown by De Haas et al. [2] that certain flavours of pairwise alignment can outperform geometric models of chord similarity. In this paper, we argue that integrating the crowd knowledge of chord relationships can improve pairwise alignment even more. There are a couple of factors affecting the improvement in pairwise sequence comparison methods. These consist of comparison algorithms and scoring matrices. Durbin [3] argues that meaningful, high quality alignments are solely dependent on the knowledge captured by the matrix used. Scoring matrices contain a value for each possible substitution and are used to score an alignment. Creating a matrix that assigns high scores to related sequences while penalizing unrelated ones is not a trivial task. In this paper, we propose two substitution matrices that are derived from a large amount of crowd-sourced chord sequences, with the aim of creating chord similarity models that capture the variation of user-generated chord sequences.

II. Related Work

Chord similarity has been only addressed by cognitive and mathematical studies. Bharucha and Krumhansl are two of the earliest and most prolific researchers on perceived similarity between tones, chords and keys in tonal music. In her book [4], Krumhansl nicely summarizes her listening experiments with various collaborators. Closely related to this paper is a chord similarity model [5] based on an experiment during which all possible pairs of diatonic triads were judged in terms of how well they sounded in succession.

One of the most famous and elaborate models was developed by Fred Lerdahl [6]; the Tonal Pitch Space (TPS) model is a mathematical approximation of the perceived relations between chords. TPS was proven to correspond well to the findings of Bharucha and Krumhansl [7], [8] and more suitable for calculating chords distances within same key contexts [9].

A number of mathematical models of chord similarity have been proposed over the years in the literature. Some, such as [10], use cognitive studies as a foundation to formulate a mathematical model, while others [11]–[13] are based on heuristics and expert knowledge. Rocher *et. al.* [14] presented a comparative study on a series chord similarity models in the context of chord analysis. Although the results indicate that the choice of model is dependent on the musical application, Lerdahl's TPS is usually in the top-ranked models.

Finally, although the notion of chord sequence variation has been addressed either directly [15] or via the automatic accompaniment [16] or harmonization proxies, the mutability of chords between song variants has remained rather unexplored.

III. The alignment Scoring Model

Given two sequences $x = \{x_1, x_2, .., x_n\}$ and $y = \{y_1, y_2, ..., y_m\}$ where each symbol x_i and y_i come from the same finite alphabet, we want to assign a value/score to their alignment. This score should represent the relative likelihood that the two sequences are related as opposed to being unrelated (aligned by chance). This is typically modeled by a ratio, denoted as odds ratio: $P(x, y|M)/P(x, y|R)$, where M is the match (related) model and R is the random (unrelated) model.

If q_a the probability of a symbol a, then for the random alignment case, aligned pairs happen independently, which translates to:

$$P(x, y|R) = \prod_i q_{x_i} \prod_j q_{y_j} \tag{1}$$

For the matching case, where aligned pairs happen with a joint probability p, the probability for the alignment is:

$$P(x, y|M) = \prod_i p_{x_i y_i} \tag{2}$$

In order to get an additive scoring system, it is standard practice to get the log of (1), which after substituion becomes:

$$log \frac{P(x, y|M)}{P(x, y|R)} = \sum_i log\Big(\frac{p_{x_i y_i}}{q_{x_i} q_{y_i}}\Big) \tag{3}$$

A substitution or scoring matrix is nothing more than a matrix arrangement of the $log(p_{x_i y_i}/q_{x_i} q_{y_i})$ values (scores) of all possible pairwise symbol combinations.

The major difficulty of the scoring matrix calculation is the computation of the joint probability $p_{x_i y_i}$ that expresses the likelihood of symbols x_i and y_i to be related. The key idea for solving this problem is that trusted alignments of related sequences can provide information regarding the mutability of symbols.

IV. Deriving score parameters from Chordify edits

Each *Chordify* transcribed song is automatically assigned a meter and divided into bars while aligned to the corresponding audio. Users can insert, delete, replace, and shift the automatically estimated chords. The fact that two or more users might have edited the same chord position with different chords allows us to speculate that certain chords are more likely to be confused than others. Our goal is to capture that chord confusion.

We gathered 671 edits from 46 *Chordify* songs with 14.5 (± 26.8) edited sequences per song. For the sake of simplicity we normalize each sequence to the same key, and convert all chords to their closest major or minor versions (*majmin* representation). At the next step, each sequence was normalized to the estimated key and mapped to a 24-size alphabet $\mathcal{A} = \{A, B, C, ..., X\}$ to account for all chromatic root notes. The final substitution matrix, which we call Confusion Substitution Matrix (**CSM**), was generated using the `SubsMat` package from the bioinformatics library `Biopython` with the default settings.

V. Evaluation of the Confusion Substitution Matrix

In the following sections we aim to evaluate the strength of the **CSM**. Classification and retrieval on real life data can act as proxies of alignment quality. We will firstly present the competing, alternative scoring systems.

A. Alternative scoring schemes

The most basic alignment scoring scheme assigns a +1 score to matching pairs and -1 otherwise. The resulting substitution matrix would have -1 in all its cells except from those on the diagonal. We denote this matrix **Exact** since it promotes only exact matches.

We generate a substitution matrix, denoted **Edit**, based on the chord similarity scheme of Macrae and Dixon [17]. This scheme has been used successfully [17] on crowd-sourced transcriptions similar to our test dataset (see section 5.2). It has not been reviewed by Rocher *et. al.* [14] thus it is worth putting it to the test.

As we have previously mentioned, **Lerdahl** introduced a model of Tonal Pitch Space in [6]. This model has been studied [8], used as a basis for other models [2], while it has been proven to be very powerful in various contexts [14]. We use the proposed chord distance measure to create a similarity matrix by taking the complement of the normalized distance in the range of [0, 1]. The chord relationships that are absent in the original model (e.g. A and A# in the key of A) are assigned a similarity score of -1.

Finally, we include an alternative cognitive model of chord relations from an 1983 experiment of Bharucha and Krumhansl [5], during which all possible pairs of diatonic triads were judged in terms of how well they sounded in succession. The **Bharucha** model has been compared to Lerdahl's in [8] as they both represent relations between the same objects. It should be noted, that the same process as in **Lerdahl** was used to convert the original distance matrix into a similarity matrix.

B. Test dataset

From *UltimateGuitar* we web-mined 948 user chord transcriptions corresponding to the 174 songs of the complete Beatles discography as provided by the famous Beatles dataset[1]. The mean of the number of transcriptions per song is 5.66 (±3.61). Selecting this artist was based on the following: (a) Beatles' popularity translates to large number of user-transcription variations, (b) the Beatles dataset has been used extensively to train and test ACE systems.

C. Classification & Retrieval tasks

The scoring models are compared on two different tasks, classification and retrieval. Regarding the classification scenario, each chord sequence in the test dataset belongs a "family" that is the song it aimed to transcribe. We are interested in examining which scoring matrix can better predict the family of an unknown chord sequence. From each of 27 songs with more than 9 transcriptions, we randomly select 8 transcriptions. Therefore we create a subset of the original test set that contains 216 transcriptions grouped into 27 families. For every transcription in the set we perform a multiclass k-Nearest Neighbor (kNN) classification experiment. Such an experiment, in contrast to binary classification, allows for the investigation of cases where $k > 8$. It also complements the retrieval task by probing the ranking strength of each model. In order to keep the number of transcriptions per family balanced, each query is taken out of the test set and replaced with a random transcription from the same family (that is not already in the set).

In the retrieval scenario we create a list of transcriptions ranked by relevance for each chord sequence in the dataset. We measure the Average Precision (AP). We want to examine whether the mean AP difference between the scoring matrices is statistically significantly different to zero.

VI. Results

Classification results for various numbers of neighbors are presented in Table I. Results of significance testing between the two most accurate matrices for each k using the *t-test* are also presented. We observe that the **CSM** matrix shows superior performance for $k > 20$ and almost indistinguishable performance from the top-performing matrix for $k < 20$. We argue that for very small k values, the accuracy is biased towards exact matching models. The discussion in the retrieval task results will shed more light on this matter.

Regarding the retrieval task, we first observe that the AP scores for all matrices do not follow a Gaussian distribution. We therefore performed a Friedman test to assess the significance of the AP performance differences between matrices. We found a statistically significant difference in AP depending on the matrix with $p < 10^{-15}$. Median AP scores for the **CSM**, **Edit**, **Exact**, **Lerdahl** and **Bharucha** trials were 0.68 ± 0.31, 0.67 ± 0.31, 0.67 ± 0.31 and 0.66 ± 0.31 respectively. Post hoc analysis with Wilcoxon-Nemenyi-McDonald-Thompson test was conducted to identify which pairs of matrices exhibit significantly different behaviour. We found significant differences for all pairs with $10^{-20} < p < 10^{-4}$ except **CSM** and **Exact**. The indistinguishable performance between **CSM** and **Exact** comes as no surprise; the formulation of average precision and consequently the mean of AP are dominated by "easy" queries, which are abundant in our test set.

k	*CSM*	*Edit*	*Exact*	*Lerdahl*	*Bharucha*	Signif.
1	87.0 (1.6)	87.0 (1.4)	86.0 (1.3)	85.0 (1.4)	85.0 (1.0)	
2	79.8 (1.3)	80.5 (1.5)	80.0 (1.3)	79.9 (1.5)	79.3 (1.3)	
3	75.0 (1.5)	76.0 (1.4)	75.5 (1.4)	75.0 (1.9)	75.0 (1.5)	
4	75.0 (2.3)	76.5 (1.5)	77.0 (1.9)	75.0 (2.2)	76.0 (1.9)	
5	77.0 (2.2)	**78.8** (1.9)	77.5 (1.9)	75.0 (2.2)	76.5 (1.9)	*
6	75.5 (1.8)	76.0 (2.5)	75.5 (2.2)	73.5 (2.6)	74.5 (2.5)	
7	73.5 (2.0)	74.0 (2.3)	73.5 (2.1)	71.0 (2.6)	72.5 (2.5)	
8	72.1 (1.7)	**74.5** (1.9)	72.5 (1.7)	71.5 (2.0)	72.5 (1.9)	***
9	73.8 (1.7)	75.0 (1.8)	73.5 (1.2)	72.5 (1.7)	73.0 (1.7)	
10	71.5 (1.4)	**72.0** (1.8)	70.0 (1.4)	71.5 (1.7)	70.2 (1.6)	*
20	**62.5** (1.2)	59.5 (1.7)	59.5 (1.5)	57.0 (1.9)	58.0 (1.9)	***
25	**59.5** (1.1)	53.5 (1.4)	52.5 (1.2)	53.5 (2.2)	54.2 (1.7)	***
30	**54.5** (1.8)	49.5 (1.2)	50.8 (1.5)	50.0 (1.9)	52.5 (1.8)	***
35	**52.5** (1.6)	47.5 (1.2)	46.8 (1.7)	48.5 (2.2)	49.0 (1.9)	***
40	**49.5** (1.8)	45.5 (1.6)	43.5 (1.8)	47.0 (2.5)	47.0 (2.2)	***
45	**45.5** (2.3)	42.4 (1.7)	41.5 (1.3)	42.0 (1.5)	44.0 (2.0)	**
50	**42.0** (1.7)	38.9 (2.3)	39.0 (1.0)	39.5 (1.8)	40.5 (1.8)	*
55	**39.4** (1.6)	35.0 (2.5)	37.0 (1.5)	37.0 (1.8)	38.0 (1.7)	***
60	**36.5** (1.7)	32.6 (2.1)	35.5 (1.3)	35.5 (2.2)	34.5 (1.8)	*

TABLE I

CLASSIFICATION ACCURACY MEDIAN (STD) FOR k IN THE RANGE OF 1 TO 60. SIGNIFICANCE BETWEEN THE TWO MOST ACCURATE MATRICES AT EACH k IS REPRESENTED AS NUMBER OF "*" ($p < 10^{-3}$:"*", $p < 10^{-4}$:"**", $p < 10^{-5}$:"***").

It is worth considering an evaluation that sheds light on the bad performing queries and is focused on the low-end of the AP spectrum: the Geometric Mean Average Precision (GMAP) [18]. We run the retrieval experiment, and computed the GMAP for **CSM** and **Exact**, 20 times on randomly generated subsets of the query set. Each subset had a size 40% of the original set. The resulting GMAP distributions do not meet normality assumptions, therefore we performed a Wilcoxon signed-rank test to assess whether their population mean ranks differ. We found a significance difference between **CSM** (0.478 ± 0.0019) and **Exact** (0.461 ± 0.0020) with $p < 10^{-5}$. This finding, combined with the AP results, proves that our model and exact matching present almost identical strength in ranking easy queries on the top; however, **CSM** manages to rank higher those chord transcriptions of high variation. This is also supported by the classification results for higher k values.

VII. Deriving Score Parameters from Cover Transcriptions

In the following sections we focus on the mutability of chords in between variations of the same song (e.g. cover songs). We want capture the way artists vary certain chords in order keep the original song interesting yet recognizable. It should be noted that there is a fundamental difference between the two models. **CSM** aims to capture the results of the unconscious and noisy process of identifying a chord. A chord mutability model on the other hand, should capture the results of the conscious and informed decisions to change one chord for another.

[1] isophonics.net/content/reference-annotations-beatles

From a list of most covered artists[2] we firstly selected the top thirty. At the next step, we searched for "Artist+Song title" and "Song title" and web mined the results from *UltimateGuitar.com*. We managed to gather (more than one) covers for 148 songs. Similar to the previous section we converted each chord transcription to its *majmin* representation.

In contrast to the *Chordify* edits, *UtlimateGuitar* cover transcriptions are not aligned to the original. However, one of the fundamental prerequisites for building a substitution matrix is trusted alignments. In order to solve this, we employ the only thing in common between the original and the covers; the lyrics. *UltimateGuitar* employs an unwritten formatting rule that aligns chords to lyrics. For example, in the transcription of "Let It Be" below, the C chord should be played when the song reaches the word "find" and the G chord when it reaches "times" and so on.

```
      C                 G                   Am
When I find myself in times of trouble, Mother Mary ...
```

One would expect lyrics should remain almost intact in a cover. In practice however, cover lyrics can be noisy. In order to solve these issues we have devised a process of chord alignment via lyrics alignment comprised of three steps. In the first step we perform pairwise, word-based alignment on the two different versions of lyrics. Based on the introduced gaps "-" in the lyrics, we shift the corresponding chords for each transcription accordingly. In the second step we locate those sections of lyrics that are identical between the aligned versions. We assume that only chords inside exact matching sections can be matched with high confidence. Finally, we locate all the matching aligned chord pairs.

VIII. Evaluation & Results for the Mutability Substitution Matrix

The substitution matrix generated from the aligned chord pairs is called Mutability Substitution Matrix (**MSM**). We are interested in our model's strength with regard to cover detection. In a retrieval scenario given a query song, our model should rank higher than **Exact** those songs that are covers of the query. In order to ensure that our model captures the actual chord mutability rather than an artifact of frequency of chords' appearances, we also compare it to a substitution matrix based on an artificial alignment; each aligned pair of chords $(c_i, c_j) \in \mathcal{A} = \{A, B, C, ..., X\}$ appears with joint probability $P(c_i)P(c_j)$, where P the probability of an individual chord appearing in the covers dataset. We call this matrix **Probabilistic**.

We performed a Shapiro-Wilk normality test on that MAP score distributions and found that only **MSM** is normally distributed. Consequently, we performed a Wilcoxon signed rank and found significance difference between **MSM** (0.584 ± 0.061) and **Exact** (0.539 ± 0.058) with $p < 10^{-8}$. For the sake of completeness, we also compared the accuracies of both models in a 1-NN classification scenario. Once again we found significance difference between **MSM** (0.623 ± 0.067) and **Exact** (0.588 ± 0.059) with $p < 10^{-5}$. The much lower MAP and accuracy scores for **Probabilistic**, 0.11 and 0.05 respectively, strongly indicate that our model captures something more meaningful than a random artifact of the probability of chords appearing in a song.

IX. Conclusions

We presented two novel data-driven models of chord similarity and mutability based user-data from *Chordify* and *UltimateGuitar*. Using user generated chord sequences, we built a substitution matrix for pairwise alignment that significantly outperforms heuristic chord similarity models in classification and retrieval tasks. Using user-transcriptions of cover songs we proposed a data-driven substitution matrix that outperforms exact matching methods in a cover detection scenario. We argue that our models capture the collective perception of chord relations in user-generated content.

References

[1] B. de Haas, J. P. Magalhaes, R. C. Veltkamp, and F. Wiering, "Harmtrace: improving harmonic similarity estimation using functional harmony analysis." in *Proceedings of the International Society for Music Information Retrieval Conference*, 2011, pp. 67–72.

[2] B. de Haas, F. Wiering, and R. C. Veltkamp, "A geometrical distance measure for determining the similarity of musical harmony," *International Journal of Multimedia Information Retrieval*, vol. 2, no. 3, pp. 189–202, 2013.

[3] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[4] C. L. Krumhansl, *Cognitive foundations of musical pitch*. Oxford University Press New York, 1990, vol. 17.

[5] J. Bharucha and C. L. Krumhansl, "The representation of harmonic structure in music: Hierarchies of stability as a function of context," *Cognition*, vol. 13, no. 1, pp. 63–102, 1983.

[6] F. Lerdahl, *Tonal pitch space*. Oxford University Press, 2001.

[7] R. Randall and B. Khan, "Similarity measures for tonal models," in *Proceedings of the International Conference on Music Perception and Cognition*, 2006.

[8] R. R. Randall and B. Khan, "Lerdahl's tonal pitch space model and associated metric spaces," *Journal of Mathematics and Music*, vol. 4, no. 3, pp. 121–131, 2010.

[9] E. Bigand and R. Parncutt, "Perceiving musical tension in long chord sequences," *Psychological Research*, vol. 62, no. 4, pp. 237–254, 1999.

[10] M. Yoo and I. Lee, "Musical tension curves and its applications," in *Proceeding of International Computer Music Conference*, 2006.

[11] E. Chew, "Towards a mathematical model of tonality," Ph.D. dissertation, 2000.

[12] B. Mathieu, "Outils informatiques d'analyse musicale," *Master's thesis, ENST-Bretagne*, 2002.

[13] J. F. Paiement, D. Eck, and S. Bengio, "A probabilistic model for chord progressions," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2005, pp. 312–319.

[14] T. Rocher, M. Robine, P. Hanna, and M. Desainte-Catherine, "A survey of chord distances with comparison for chord analysis," in *International Computer Music Conference*, 2010.

[15] J. Garbers, A. Volk, P. van Kranenburg, F. Wiering, L. P. Grijp, and R. C. Veltkamp, "On pitch and chord stability in folk song variation retrieval," in *Mathematics and Computation in Music*. Springer, 2009, pp. 97–106.

[16] H. V. Koops, J. P. Magalhaes, and B. De Haas, "A functional approach to automatic melody harmonisation," in *Proceedings of the workshop on Functional art, music, modeling & design*, 2013, pp. 47–58.

[17] R. Macrae and S. Dixon, "Guitar tab mining, analysis and ranking." in *Proceedings of the International Society on Music Information Retrieval conference*, 2011, pp. 453–458.

[18] S. Robertson, "On gmap: and other transformations," in *Proceedings of the conference on Information and knowledge management*, 2006, pp. 78–83.

[2] www.whosampled.com/most-covered-artists