MODELING HARMONIC SIMILARITY FOR JAZZ USING CO-OCCURRENCE VECTORS AND THE MEMBRANE AREA

Carey Bunks[1,2] **Tillman Weyde**[2] **Simon Dixon**[1] **Bruno Di Giorgi**[3]
[1] Queen Mary University of London, UK
[2] City, University of London, UK
[3] Apple, UK

ABSTRACT

In jazz, measuring harmonic similarity is complicated by the common practice of reharmonization – the altering or substitution of chords without fundamentally changing the piece's harmonic identity. This is analogous to natural language processing tasks where synonymous terms can be used interchangeably without significantly modifying the meaning of a text. Our approach to modeling harmonic similarity borrows from NLP techniques, such as distributional semantics, by embedding chords into a vector space using a co-occurrence matrix. We show that the method can robustly detect harmonic similarity between songs, even when reharmonized. The co-occurrence matrix is computed from a corpus of symbolic jazz-chord progressions, and the result is a map from chords into vectors. A song's harmony can then be represented as a piecewise-linear path constructed from the cumulative sum of its chord vectors. For any two songs, their harmonic similarity can be measured as the minimal surface membrane area between their vector paths. Using a dataset of jazz contrafacts, we show that our approach reduces the median rank of matches from 318 to 18 compared to a baseline approach using pitch class vectors.

1. INTRODUCTION

Measuring similarity between songs is important for many music information retrieval tasks, for example, recommendation systems, copyright infringement detection, and genre classification systems. Many different types of features can be used to compare songs, but the specific focus of this paper is on jazz harmony as represented by the symbolic chord progressions found on leadsheets.

The analysis of harmonic similarity has been studied using N-grams [1], parse trees [2, 3], and NLP methods such as TF-IDF, Latent Semantic Analysis (LSA), and Doc2Vec [4]. The approach taken in this paper is based on embedding chord symbols into a vector space through the computation of a co-occurrence matrix [5]. As will be seen when we describe the data in Section 2, many chord symbols occur only rarely. To reduce computational problems due to sparsity, the dimensionality of chord space should be reduced [6]. A typical machine learning approach for this might use an algorithm such as truncated singular value decomposition after vectorization [7]. In this work, however, we use music theory to reduce the number of effective chord symbols prior to vectorization, which in turn reduces the chord space dimensionality. In the ensuing sections we describe the data, explain our approach to dimensionality reduction, and give computational details of how we compute the co-occurrence matrix. We then explain how the chord vectors generated from the co-occurrence matrix are used to represent chord progressions, and we present a novel harmonic-similarity metric, the *membrane area*.

The experimental part of our paper is based on analyzing contrafacts. In jazz, a contrafact is a song whose harmony is similar to that of another song, but which has a different melody [8]. The tune *I Got Rhythm*, by George Gershwin (1930), is a well-known source of many contrafacts, [1] and there are numerous other examples [9–11]. In addition to the difference in melody, contrafact chord progressions often feature reharmonization, a common practice in jazz that makes chord substitutions in a song while maintaining its harmonic identity [12]. Reharmonization is a core characteristic of jazz – so much so that there are typically reharmonizations from chorus to chorus even in a single performance of a jazz song.

2. THE DATA

The data used in this paper is a corpus of symbolic chord progressions similar to those found in jazz fake books, such as the Real Book [13]. The progressions are mainly from jazz standards, but also include some blues, jazz-blues, modal jazz, and jazz versions of pop tunes. The corpus is derived from a collection distributed with *Impro-Visor*, an open-source music notation program. [2] Our modifications remove control information used by the Impro-Visor application, retaining the musical content and song-specific metadata. We have performed numerous quality checks on the data, have made corrections where required, and have enriched some of the metadata. The re-

[1] `https://en.wikipedia.org/wiki/Rhythm_changes`
[2] `https://www.cs.hmc.edu/~keller/jazz/improvisor/`

© C. Bunks, T. Weyde, S. Dixon, and B. Di Giorgi. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** C. Bunks, T. Weyde, S. Dixon, and B. Di Giorgi, "Modeling Harmonic Similarity for Jazz Using Co-occurrence Vectors and the Membrane Area", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

sulting corpus and the code we used to generate our examples is available on GitHub.[3] The Impro-Visor corpus provides chord progressions for 2,612 songs, and is the largest digital collection of jazz chord progressions we know of. For comparison, the applications iRealPro[4] and Band-in-a-Box[5] contain chord progressions for roughly 1400 and 226 jazz standards, respectively. The Weimar Jazz Database contains chords for 456 jazz songs.[6]

Of the 134,182 chord symbol instances in the corpus, there are 1,542 unique symbols, of which many are rare, with 20% occurring just once, and 50% fewer than six times. As the corpus consists mainly of jazz standards, there is a preponderance of 7^{th} chords, comprising at least the root, 3^{rd}, 5^{th}, and 7^{th} notes. These types of chords often have additional extensions (9^{th}, 11^{th}, 13^{th}) and chromatic alterations (♭9, ♯9, ♭5, ♯5). A common variation of jazz chords replaces the 7^{th} with a 6^{th} for major7 and minor7 chords. As 7^{th} chords are the basic harmonic unit in jazz [14], and make up 77% of our corpus, they are the focus of our approach to dimensionality reduction described in the next section. Of the remaining chords, 16% are three-note chords (triads), and 7% are drawn from a variety of special types, as shown in Table 1, which provides a list of all the types and their frequencies.

Type	Percentage
7^{th} chords (and extensions)	76.939%
major triads	11.484%
slash chords	4.781%
minor triads	4.320%
sus chords	1.364%
no chord	0.458%
augmented triads	0.392%
major triads add9	0.127%
diminished triads	0.095%
power chords	0.031%
polychords	0.009%

Table 1. Corpus chord types and their frequencies

3. DIMENSIONALITY REDUCTION

Our approach to reducing dimensionality is based on mapping chords to a reduced vocabulary of functionally equivalent symbols (similar to [15]). This is important because 20% of the chords in the corpus occur only a single time (known as *hapax legomena*), and without additional processing, these types of terms would provide no predictive value [16]. Many techniques are used in NLP to better leverage hapax legomena. For example, stemming, lemmatization, and thesauri are all useful. This paper takes a similar approach for harmony, making use of music theory to reduce the dimensionality of chord space. Our method is akin to lemmatization, applying concepts from functional harmony to group similar chords into classes (for example, see [17]). Based on standard practices in jazz [12, 18, 19], we reduce the set of 1,542 chord symbols to 61 chord classes, as detailed in the following sections.

3.1 7^{th} Chord Types

Our choice of base chord types is built on the four-note 7^{th} chords diatonically generated from the major scale, and making up 77% of our corpus. These are the major7 (M), minor7 (m), dominant7 (7), and minor7♭5 (h), where the symbols shown in parentheses are abbreviations we use in this paper. To these we add a fifth base chord type, the diminished7 (o). Combining the five types with the root notes from the 12 pitch classes yields 60 chord classes. Instances of these classes can occur with extensions or alterations, and we map these to the base class without extension/alteration. For example, we map the symbols Cm9 and Cm11 to the Cm7 class; C7♭9, C7♯5, and C13 to the C7 class; and CM7♯11 to the CM7 class. In addition, in accordance with reharmonization practices, we assign chords such as CmM7 to the Cm7 class and C6 to the CM7 class. We also include the symbol *NC* (no chord) to account for the absence of harmony (0.5% of the corpus).

3.2 Other Chord Type Mappings

In the following discussion, we describe a rationale for mapping the remaining 22.5% of the symbols into classes of the five base types defined above. The mapping choices described in the following discussion are imperfect, but they are simple to implement, and we show they are adequate for our application.

3.2.1 Triads

Triads represent 16% of the corpus. As they do not contain a 7^{th} note, mapping them to the base chord types can be ambiguous. For example, a C major triad shares all of its notes with both the CM7 and C7 chords. We attempt to resolve triad ambiguities using principles from tonal harmony and the local harmonic context. Based on the chord following a triad, we decide whether it has a subdominant, dominant, or tonic function [19]. For example, for a major triad, if the root of the following chord is a fifth down and a member of the major7 or minor7 classes we assign the triad to the dominant7 class with the same root. Otherwise, we assign it to its corresponding major7 class. Major triads with an added 9^{th} are handled in the same way. Augmented triads share their notes with dominant7♯5 chords, an alteration of the dominant, and so we map these to the dominant7 class with the same root. Finally, we map all the minor and diminished triads to their corresponding minor7 and diminished7 classes, respectively.

3.2.2 Sus Chords and Slash Chords

Sus chords also have a harmonic function that depends on context [18]. When followed by a dominant7 chord with

[3] https://github.com/carey-bunks/Jazz-Chord-Progressions-Corpus
[4] https://www.irealb.com/forums/showthread.php?12753-Jazz-1350-Standards
[5] https://members.learnjazzstandards.com/sp/biab-jazzstandards/
[6] https://jazzomat.hfm-weimar.de/dbformat/dbcontent.html

the same root, they act like a subdominant and we opt to map them to a minor7 class with a root a fifth above. For example, a G7sus4 would map to a Dm7. Otherwise, they act like a dominant and we map them to the dominant7 class with the same root. Slash chords are chords played over a specific bass note, for example C/G or Dm7/G, where the symbol above (to the left of) the slash is the chord and below is the bass note. If the bass note belongs to the chord above the slash (for example, C/G), it is an inversion. For such cases, we map it to the class of the chord above the slash. Slash chords are also commonly used to represent sus chords. For example, Dm7/G is harmonically equivalent to G9sus4. We map these according to the process for sus chords. For all other slash chords, we map the chord as if the bass note were an extension or alteration of the chord above the slash.

3.2.3 Power Chords and Polychords

Power chords consist of just two notes, a root and a fifth. As they have no 3rd or 7th, they are harmonically ambiguous. With only 42 instances in our corpus, we have opted to map these chords to the no-chord class. With only 12 instances, polychords are also rare. These chords, used mainly by pianists, consist of a lower triad and an upper triad or 7th chord. We map polychords according to their lower structure, interpreting the upper structure as a collection of extensions or alterations.

4. KEY SIGNATURE BASED REPRESENTATION

To make distributional semantics more effective, we transpose all songs to a common key, and represent them in Roman numeral notation. However, transposition requires knowing the correct key of each song, and from extensive manual checking, we know that our database contains a fair number of songs for which the stated key signature is in error. For this reason, we introduce a key signature estimation algorithm, as described in the following section.

4.1 Key Signature Estimation Algorithm

Several authors have proposed key estimation algorithms for music information retrieval tasks [20–24]. However, our objective is not to estimate the key that is cognitively perceived by a listener, but rather a simpler problem, the key signature that minimizes the number of accidentals needed when writing out the song's chords. Some prior work exists for this [25], however, it is based on machine learning models applied to MIDI data for classical music. Our algorithm selects the key signature most consistent with the chord progression. For each chord in a progression, we map it to one of the described 61 classes, and identify all the major scales it could belong to (excluding diminished7 and no chord classes). The major scale that accumulates the most beats is the resulting estimate of the key signature for that song.

Figure 1 provides a concrete illustration of how the key estimation algorithm works for the case of a short chord progression: A7-Dm7-G7-CM7-CM7. Each column of the table represents one measure, and in this example, there is one chord per measure. The column labels correspond to the chords, and each row label is a key signature whose major scale diatonically contains one or more of the chords in the progression. As shown, the A7 chord belongs to D major; the Dm7 chord belongs to B♭, C, and F major; G7 belongs to C major; and CM7 belongs to both C and G major. Presuming four beats per measure, C accumulates the most beats (16), and is the resulting key signature estimate.

Example: 6-2-5-1 Chord Progression

Major Scales	A7	Dm7	G7	CΔ7	CΔ7	Totals
Bb		4				4
C		4	4	4	4	16
D	4					4
F		4				4
G				4	4	8

Figure 1. Illustration of key signature estimation

4.2 Algorithm Evaluation

As already mentioned, there are quite a few songs in our corpus where the key signature is incorrect or in doubt. Nevertheless, it is worthwhile comparing the outputs of our key estimation algorithm with the keys recorded in the corpus. Of the 2,612 songs, the algorithm concurs with the database for 1,763 (67.5%) of them. For the 849 songs with database key signatures that do not agree with our estimates, we use the Circle of Fifths as a distance metric to evaluate the magnitude of differences between the two. Adjacent key signatures on the circle of fifths correspond to major scales that differ in a single pitch class. Table 2 shows the distribution of circle-of-fifths distances between estimated and database key signatures for all of the songs in the corpus. The first row is the distance in number of sharps or flats from the estimated to the database key, where 0 corresponds to agreement. The last column of Table 2 is labelled "Amb." for ambiguous. There are 123 songs in the database for which the key estimation algorithm returns a non-unique result, finding two or more equally good major scales. This occurs for 4.7% of the songs in the corpus, and when it does our estimation algorithm defaults to the database key.

Dx	6♭	5♭	4♭	3♭	2♭	1♭	0	1♯	2♯	3♯	4♯	5♯	Amb.
Frq	10	22	33	55	99	304	1763	183	22	25	12	1	123

Table 2. Key signature estimation statistics with the circle of fifths distance **Dx** by the frequency of occurrence **Frq**

4.3 Mapping to Roman Numeral Notation

Once a song's key has been estimated, all the chords in its progression can be mapped to Roman numeral notation. Table 3 shows the Roman numerals corresponding to chord roots for C major. As an example, the sequence of chords A7-Dm-G7-CM maps to vi7-iim-v7-iM. In our system, we represent minor keys by their relative major, so the relative minor cadence, Bm7b5-E7-Am7, maps to viih-iii7-vim.

Root	C	D♭	D	E♭	E	F	G♭	G	A♭	A	B♭	B
RN	i	♭ii	ii	♭iii	iii	iv	♭v	v	♭vi	vi	♭vii	vii

Table 3. Roman numeral notation: chord roots in C major

5. VECTOR REPRESENTATION

Sections 3 and 4 described our approach for reducing the dimensionality of chord space, distilling the 1,542 chord symbols in our corpus to 61 classes. In this section we describe our method for embedding the chord classes into a vector space. Our design objective is that common reharmonizations be close to each other in cosine similarity, and it is known that the co-occurrence matrix can capture this type of characteristic [5, 26–28].

Given a corpus of D chord progressions, with progression $d \in \{1, 2, \ldots, D\}$ containing N_d chords with indices $1, 2 \ldots, N_d$, we can represent the corresponding sequence of chord symbols as $s_{d,1}, s_{d,2}, \ldots, s_{d,N_d}$. We define the symmetric, sliding context window, $W_{k,d}$, of nominal width N_w with the indices $W_{k,d} = [w_l, \ldots, (k-1), (k+1), \ldots, w_r]$, where the left and right endpoints are $w_l = \max(k - N_w, 1)$ and $w_r = \min(k + N_w, N_d)$, respectively. With these definitions, the $(i, j)^{\text{th}}$ element of the co-occurrence matrix, $C_{i,j}$ is computed by

$$\mathbf{C}_{i,j} = \sum_{d=1}^{D} \sum_{k=1}^{N_d} \sum_{w \in W_{k,d}} \begin{cases} 1, & \text{if } s_{d,k} = c_i \text{ and } s_{d,w} = c_j \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

This produces a square, symmetric matrix whose row $\mathbf{C}_i$ (or alternatively, column) is a vector representations of the i^{th} chord class c_i. As it will be useful in the following, we normalize each row to have unit length. Because co-occurrence matrices capture contextual information, the vectors of chord classes that have similar harmonic function are expected to be close to each other with respect to the cosine similarity measure, and this seems to be borne out by an inspection of certain chord vectors. For example, of 60 chord classes, the closest vector to the v7 is its tritone substitute, the ♭ii7, and the closest to the iim is the iih, a common substitute from the parallel minor scale (see modal interchange in [19]).

6. MEMBRANE-AREA DISTANCE METRIC

We use the co-occurrence vectors to represent chord progressions in a way that represents each chord type, duration, and metric position, while being robust to reharmonizations. The normalized chord vectors derived from the co-occurrence matrix can be used to plot the path of a song's progression through 61-dimensional space. Starting from the origin, the sequence of chord vectors can be concatenated from head to tail, beginning with the first, and terminating with the last vector (see Figure 2). Each unit vector is scaled by the number of beats of the chord it represents, and the result is a piecewise linear function through $\mathbf{R}^{61}$. The comparison of two songs in this space can be formulated as a trajectory comparison problem, for which there are many existing techniques [29]. The most popular ones, however, are not well adapted to our problem. The Fréchet distance, dynamic time warping, longest common subsequence, and the edit distance are all based on matching and comparing points, and would not directly factor in information about reharmonized chords embodied in the co-occurrence vectors. For this reason, we introduce a new metric that accounts for reharmonizations by computing the membrane area between the paths of two songs.

Expressed formally, we represent song vector paths by piecewise linear functions of the form $\boldsymbol{f}(t) \in \mathbf{R}^{61}$, where $t \in [0, 1]$ is a parametric variable representing the number of normalized beats traversed in the song. We can move along the entire length of $\boldsymbol{f}$ in discrete, equal increments, dt, where the starting point of the function, $\boldsymbol{f}(0)$ at $t = 0$ is the origin, and the end point of the function is at $t = 1$. Given two songs and their corresponding piecewise linear functions, $\boldsymbol{f}(t)$ and $\boldsymbol{g}(t)$, and letting $K = 1/dt$, we can define a distance metric between them as the area of a 2D membrane, M, stretched between the two paths. M is calculated as the integral obtained in the limit of

$$M(\boldsymbol{f}, \boldsymbol{g}) = \lim_{dt \to 0} \sum_{k=0}^{K} \|\boldsymbol{f}(kdt) - \boldsymbol{g}(kdt)\| dt, \tag{2}$$

where $\|\cdot\|$ is the Euclidean norm. The piecewise linear functions for two identical chord progressions would, naturally, overlay each other, yielding a membrane area of zero. Two harmonically similar songs should trace out similar paths keeping the membrane area small. For example, two chord progressions that differ in just a tritone substitution will only slightly perturb the path and the membrane area between songs. Figure 2 is a notional illustration of how the measure in Equation 2 is evaluated. The red and blue paths represent two different songs, each having three chords. Each song begins at the origin, and the chord vectors are added head-to-tail to trace out a piecewise linear path. The membrane area metric is approximated by summing the lengths of the N equally spaced black line segments drawn between the two songs. Note that this way of representing the harmony of a song accounts for positions and durations of each chord in the progression, as well as capturing harmonic similarities of chord transitions.

7. EXPERIMENTS

We have designed some experiments based on a set of jazz contrafacts listed in a Wikipedia article. [7] The list has 252 jazz songs whose harmonies are known to be based on other songs (see also [30]). A subset of 91 contrafacts are available in our corpus, but for 11 of them, only a section of the harmony is borrowed, and we remove these from the list. The basic structure of all of our experiments is the same: for each contrafact, we compute the membrane area

[7] https://en.wikipedia.org/wiki/List_of_jazz_contrafacts

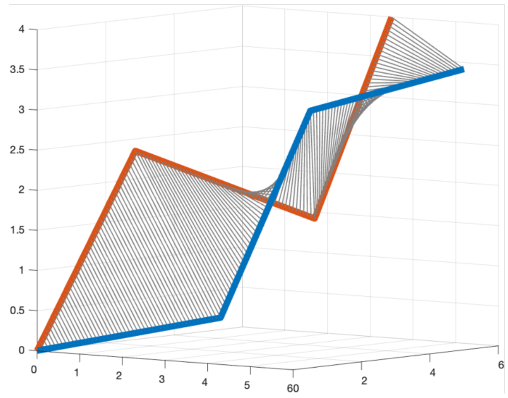

Figure 2. Conceptual illustration of the membrane-area distance metric for two, 3-chord sequences

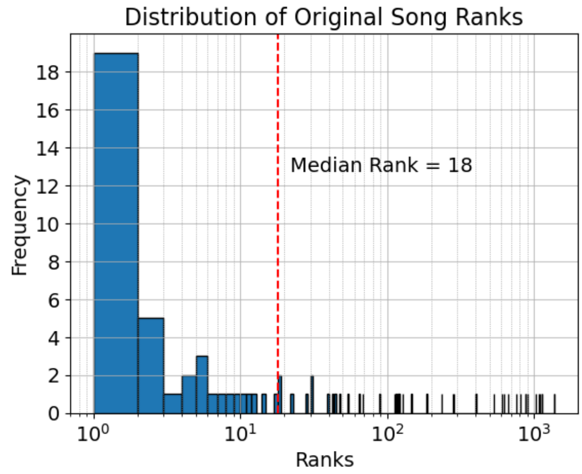

Figure 3. Histogram of original song ranks for 80 contrafacts (median rank = 18)

distance between it and each of the other 2,611 songs in our corpus. We then sort the songs from smallest membrane area to largest, and note the original song's rank in that list. Because of reharmonizations, we don't expect the membrane area to be zero for all contrafact-original pairs, but matches should rank high in the list. Original songs often inspire multiple contrafacts, and some may be closer to each other than to the original. For these reasons, we use the histogram of original song rankings to present the overall performance of our method, and we use the median rank as a method of comparison between approaches.

7.1 Using Co-Occurrence Vectors

We evaluated six variants of our approach using co-occurrence chord vectors. The first three were based on the context window widths $N_w = [1, 2, 3]$. The second three variants used the same context window values, but applied to a filtered version of the chord progressions. For each chord progression, the filter collapses adjacent identical chords to a single instance. For $N_w = 1$, this has the effect of eliminating the co-occurrence of chords with themselves, making the diagonal of the co-occurrence matrix zero. Of the six versions, the best result was obtained for the filtered chord progressions with the context window width $N_w = 1$. Figure 3 shows the histogram of original song rankings for this case. The median rank is 18, meaning that half of the original songs rank in the top 0.7% in harmonic similarity to their contrafacts. As there is some histogram mass out to rank 1,382, the histogram makes use of a log-scale on the x-axis. It is likely that some of the songs ranking better than the original are also contrafacts, as the Wikipedia list is far from exhaustive, but it would require substantial effort and expertise to evaluate this.

As noted, some original songs have inspired many contrafacts. As an example of this in our corpus, there are four known contrafacts of the song *All the Things You Are*. The ranks and membrane areas of the original song for each contrafact are shown in Table 4. The original ranks highly for three of the four contrafacts in the table. As the chord progressions for *Prince Albert* and *All the Things You are* are identical, their membrane area is zero. The contrafacts *Ablution* and *Boston Bernie* have some chord substitutions, and the original song ranks highly for both of them. The song *I Want More*, however, does quite poorly, with a rank of 758th out of the 2,611 songs in our corpus.

Contrafact	Rank	Membrane Area
Prince Albert	1	0.00
Ablution	1	6.72
Boston Bernie	2	7.72
I Want More	758	26.89

Table 4. Rank and membrane area for *All the Things You Are* against its four contrafacts

To investigate, we use the jazz harmony visualization tool described in [31] to display the chord progressions for these two songs. The visualization shows a tabular format with each rectangle representing a measure. Figures 4 and 5 show *All the Things You Are* and *I Want More*, respectively. The background colors indicate the key the chords belong to. Red is for the main key, which is A♭ for both songs. Other colors indicate modulations. Some chords are embedded in a geometric shape to indicate they are tonicizations: diamonds are secondary dominants, pentagons are borrowed chords. As the figures illustrate, the two songs have some similar chords, however, the sequences of modulations are completely different. Whereas *All the Things You Are* modulates through the tonal centers of C major, E♭ major, G major, and E major, *I Want More* modulates to D♭ major and C minor. After verifying the latter's chord progression, [8] we conclude that, harmonically, these two songs have very little in common, and we question the annotation of this song as a contrafact.

7.2 Using Pitch-Class Vectors

To evaluate the effect of using co-occurrence vectors, we compare with a baseline vector embedding scheme based

[8] Jamey Aebersold play-along book, volume 82, Dexter Gordon

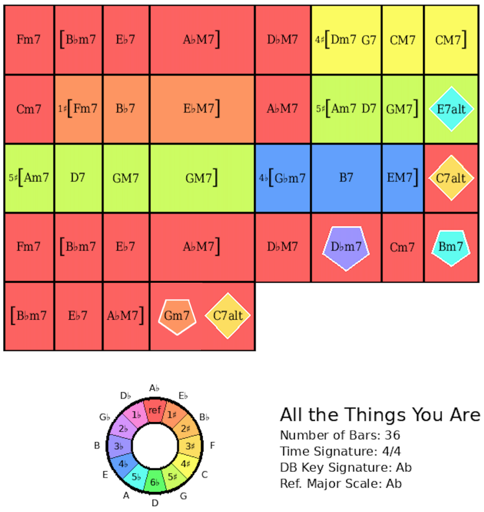

Figure 4. Chord Progression for *All the Things You Are*

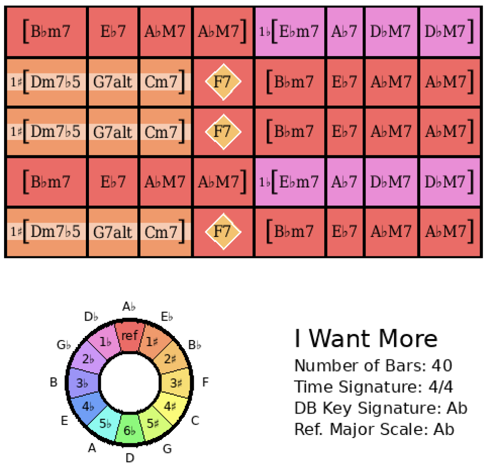

Figure 5. Chord progression for *I Want More*

on converting chord symbols to their pitch-class vectors. This is similar to the starting point of the approach used in [32]. We begin by applying the key estimation algorithm described in Section 4.1 to transpose all chords in our corpus to the key of C. Subsequently, each chord in the corpus is converted to a 12-dimensional binary pitch-class vector, with ones in positions corresponding to pitch classes belonging to the chord, and zeroes elsewhere. Thus, for a C7 chord with the notes C, E, G, and B♭, the corresponding pitch-class vector is $[1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0]$.

Following a similar schema as for the previous experiment, the pitch-class vectors can be used to construct piecewise linear paths, however, now they are constructed in a 12-dimensional space. We use the membrane area as previously to rank songs by harmonic similarity. Table 5 compares the performance of co-occurrence vectors for the best case (chord progression filtering with a window size of $N_w = 1$) versus pitch-class vectors using three metrics: median rank, mean rank, and mean reciprocal rank. Co-occurrence vectors outperform the pitch-class vectors by a large margin for each of these criteria.

Vector Type	Median	Mean	MRR
Co-occurrence	18	222	0.305
Pitch-class	318	457	0.200

Table 5. Comparison of median rank, mean rank, and mean reciprocal rank (MRR) for the filtered-progression, co-occurrence vectors ($N_w = 1$) and pitch-class vectors

8. DISCUSSION AND CONCLUSIONS

We showed how co-occurrence vectors can be used to model harmonic similarity, and introduced the membrane area as a evaluation metric that is well-adapted for handling reharmonizations. We use music theory to reduce the dimensionality of chord space, and provide a comprehensive map of all 1,542 chord symbols in our corpus to 61 classes. The results are used to compute a dense co-occurrence matrix without needing to resort to non-parametric approximations such as truncated SVD or gradient descent. Using the cosine similarity measure, we show that the rows of the co-occurrence matrix embody some characteristics of common reharmonizations. Using the normalized rows of the matrix as vector embeddings of chord classes, we modeled songs as piecewise linear paths in $\mathbf{R}^{61}$. A novel distance metric, the membrane area, was introduced and used as a measure of harmonic similarity between songs. We showed that the similarity metric can be used to retrieve contrafacts from a database of jazz standards, and that it performs significantly better than a baseline system using binary pitch-class vectors as chord embeddings.

Although our approach is successful for contrafact detection, there are several weaknesses that require future work. Our key detection algorithm is simple and static, despite the fact that jazz harmony exhibits many local key changes (e.g. see Figures 4 and 5). We also treat minor keys as equivalent to their relative major, which is not strictly correct. The chord mapping scheme is limited in its ability to distinguish common progressions such as triad progressions i-iv and v-i. A richer chord vocabulary or local key estimation could disambiguate such situations. Our song-level similarity assumes only minor structural differences between pieces. Modifying it to perform subsequence matching would overcome this limitation.

We believe that the methods discussed in this paper have many additional applications, such as those in evaluating harmonic complexity [33] and in musicology [34]. We intend to investigate whether our harmonic similarity measure can be used to cluster jazz songs by composer or decade of publication. Although our focus has been on jazz, chords have similar functions across much of Western tonal harmony. For this reason, we believe that this work can be adapted to other genres such as classical, rock, and pop. Furthermore, as our methods are based on capturing the distributional semantics of harmony, the approach may also be useful in discovering harmonic relationships in non-Western music genres.

9. ACKNOWLEDGEMENTS

The first author is a research student at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported by UK Research and Innovation [grant number EP/S022694/1].

10. REFERENCES

[1] D. Ponsford, G. Wiggins, and C. Mellish, "Statistical learning of harmonic movement," *Journal of New Music Research*, vol. 28, no. 2, pp. 150–177, 1999.

[2] W. B. De Haas, "Music information retrieval based on tonal harmony," Ph.D. dissertation, Utrecht University, 2012.

[3] M. Rohrmeier, "The syntax of jazz harmony: Diatonic tonality, phrase structure, and form," *Music Theory and Analysis (MTA)*, vol. 7, no. 1, pp. 1–63, 2020.

[4] D. Zahnd, "Similarity analysis of jazz tunes with vector space models," Ph.D. dissertation, Hochschule für Musik, Freiburg, 2022.

[5] K. Lund and C. Burgess, "Producing high-dimensional semantic spaces from lexical co-occurrence," *Behavior Research Methods, Instruments, & Computers*, vol. 28, no. 2, pp. 203–208, 1996.

[6] E. Bruni, N.-K. Tran, and M. Baroni, "Multimodal distributional semantics," *Journal of Artificial Intelligence Research*, vol. 49, pp. 1–47, 2014.

[7] C. O. S. Sorzano, J. Vargas, and A. P. Montano, "A survey of dimensionality reduction techniques," *arXiv preprint arXiv:1403.2877*, 2014.

[8] B. D. Kernfeld, *The New Grove Dictionary of Jazz*. Grove's Dictionaries Incorporated, 2002, vol. 2.

[9] H. Martin, *Charlie Parker, Composer*. Oxford University Press, USA, 2020.

[10] D. H. Rosenthal, *Hard Bop: Jazz and Black Music 1955-1965*. Oxford University Press, 1994.

[11] T. Owens, *Bebop: The Music and its Players*. Oxford University Press, 1996.

[12] D. Berkman, *The Jazz Harmony Book: A Course in Adding Chords to Melodies*. Sher Music Co., 2013.

[13] H. Leonard, *The Real Book*. Hal Leonard Publishing Corporation, 2016.

[14] R. Rawlins and N. E. Bahha, *Jazzology: The Encyclopedia of Jazz Theory for All Musicians*. Hal Leonard Corporation, 2005.

[15] X. Serra, "Audio-aligned jazz harmony dataset for automatic chord transcription and corpus-based research," in *in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 483–490.

[16] J. Pierrehumbert and R. Granell, "On hapax legomena and morphological productivity," in *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 2018, pp. 125–130.

[17] D. Harasim, C. Finkensiep, P. Ericson, T. J. O'Donnell, and M. Rohrmeier, "The jazz harmony treebank," in *in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 207–215.

[18] M. Levine, *The Jazz Theory Book*. Sher Music Co., 1995.

[19] J. Mulholland and T. Hojnacki, *The Berklee Book of Jazz Harmony*. Berklee Press, 2013.

[20] M. Mauch and S. Dixon, "Simultaneous estimation of chords and musical context from audio," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1280–1289, 2009.

[21] J. Pauwels and J.-P. Martens, "Combining musicological knowledge about chords and keys in a simultaneous chord and local key estimation system," *Journal of New Music Research*, vol. 43, no. 3, pp. 318–330, 2014.

[22] T. Rocher, M. Robine, P. Hanna, L. Oudre, Y. Grenier, and C. Févotte, "Concurrent estimation of chords and keys from audio," in *in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2010, pp. 141–146.

[23] K. C. Noland and M. B. Sandler, "Key estimation using a hidden Markov model," in *in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2006, pp. 121–126.

[24] E. Benetos, A. Jansson, and T. Weyde, "Improving automatic music transcription through key detection," in *Audio Engineering Society Conference*. Audio Engineering Society, 2014.

[25] F. Foscarin, N. Audebert, and R. Fournier-S'Niehotta, "PKSpell: Data-driven pitch spelling and key signature estimation," *arXiv preprint arXiv:2107.14009*, 2021.

[26] S. Bordag, "A comparison of co-occurrence and similarity measures as simulations of context," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2008, pp. 52–63.

[27] A. Globerson, G. Chechik, F. Pereira, and N. Tishby, "Euclidean embedding of co-occurrence data," *Advances in Neural Information Processing Systems*, vol. 17, 2004.

[28] L. Leydesdorff and L. Vaughan, "Co-occurrence matrices and their applications in information science: Extending ACA to the web environment," *Journal of the American Society for Information Science and Technology*, vol. 57, no. 12, pp. 1616–1628, 2006.

[29] K. Toohey and M. Duckham, “Trajectory similarity measures,” *Sigspatial Special*, vol. 7, no. 1, pp. 43–50, 2015.

[30] F. Tirro, “The silent theme tradition in jazz,” *The Musical Quarterly*, vol. 53, no. 3, pp. 313–334, 1967.

[31] C. Bunks, T. Weyde, A. Slingsby, and J. Wood, “Visualization of tonal harmony for jazz lead sheets,” in *24th EG Conference on Visualization (EuroVis) Short Papers*, 2022, pp. 109–113.

[32] S. Madjiheurem, L. Qu, and C. Walder, “Chord2vec: Learning musical chord embeddings,” in *Proceedings of the Constructive Machine Learning Workshop at 30th Conference on Neural Information Processing Systems (NeurIPS)*, 2016.

[33] B. Di Giorgi, S. Dixon, M. Zanoni, and A. Sarti, “A data-driven model of tonal chord sequence complexity,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 11, pp. 2237–2250, 2017.

[34] A. Moore, “Patterns of harmony,” *Popular Music*, vol. 11, no. 1, pp. 73–106, 1992.