Data-Driven Exploration of Musical Chord Sequences

Eric Nichols
Indiana University
Bloomington, IN
epnichol@indiana.edu

Dan Morris
Microsoft Research
Redmond, WA
dan@microsoft.com

Sumit Basu
Microsoft Research
Redmond, WA
sumitb@microsoft.com

ABSTRACT
We present data-driven methods for supporting musical creativity by capturing the statistics of a musical database. Specifically, we introduce a system that supports users in exploring the high-dimensional space of *musical chord sequences* by parameterizing the variation among chord sequences in popular music. We provide a novel user interface that exposes these learned parameters as control axes, and we propose two automatic approaches for defining these axes. One approach is based on a novel clustering procedure, the other on principal components analysis. A user study compares our approaches for defining control axes both to each other and to an approach based on manually-assigned genre labels. Results show that our automatic methods for defining control axes provide a subjectively better user experience than axes based on manual genre labeling.

Author Keywords
Creativity, music, chords, genre, PCA, clustering, transition matrix, HMMs

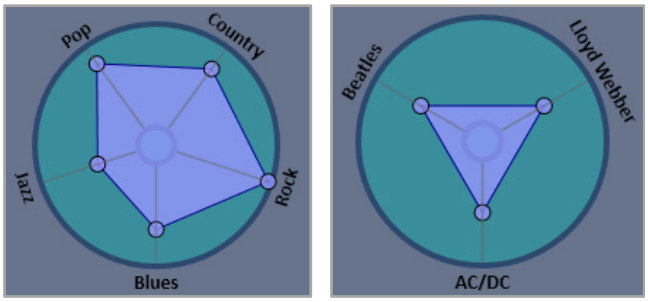
ACM Classification Keywords
H.5.5. Sound and music computing: modeling, systems.

INTRODUCTION
Artists in a variety of creative disciplines are implicitly guided by statistical principles learned through training and experience. For example, musicians develop an intuition for the "rules" of rhythm and harmony, painters learn brush strokes and color patterns, and writers acquire characteristic vocabularies and linguistic motifs. These statistical elements may be defined across entire disciplines (e.g. canonical rules of musical harmonization) or may represent an individual artist's style and technique (e.g. Monet's characteristic short brush strokes).

While computers systems will likely never equal an experienced artist's ability to shape characteristic patterns into expressive media, the use of machine learning and data-driven methods to capture the statistics that guide artists holds tremendous potential to expand computer support for creativity. Applying learned statistics may offer novices the ability to explore new artistic disciplines, may provide insight into the cognitive foundations of artistic expression, and may accelerate and diversify the work of even experienced artists by offering "intelligent scratchpads" for creative exploration.

The synthesis of musical chord sequences is an excellent domain in which to explore these possibilities. In many types of music, a *chord sequence* is an essential structural component that defines much of the identity of a piece of music. For instance, musicians frequently represent jazz and pop songs by a *lead sheet*, which describes a song as a melody, lyrics, and a chord sequence. The process of developing chord sequences is deeply creative: chords offer a powerful vocabulary for musical expression, and musicians continue to expand their mastery of chord sequences throughout years of training experience. However, this process is also loosely governed by a set of statistical rules and common patterns, including variations on those patterns that are characteristic of genres or eras of music. This domain, therefore, represents an ideal setting for exploring for data-driven support for artistic processes.

This paper thus explores data-driven exploration of musical



(a) (b)

Figure 1: Example instantiations of our polygon slider based on different parameterizations of the chord sequence space. Manipulating any of the "handles" adjusts the weight of the corresponding control axis. Applications include the generation of chord sequences based on the statistics of (a) genres or (b) artists. Axes need not be driven by underlying labels: abstract axes, such as those computed by our PCA- and randomly-seeded-clustering approaches, are also described and evaluated.

IUI'09, February 8–11, 2009, Sanibel Island, Florida, USA.

chord sequences. We present a system that enables users to generate a variety of chord sequences to accompany a melody by manipulating control axes on an intuitive graphical widget. Each axis defines the weight assigned to a learned dimension of variability in musical chord sequences. This weight-adjustment interaction is simple for novices with limited musical experience to understand, but is also useful for experts: the manipulation of weights is much quicker than manually modifying individual chords in a chord sequence, and allows experts to explore variations outside of their own characteristic chord patterns.

More specifically, using databases of music in different genres, we investigate algorithms for blending together genre-specific statistical information and for deriving new statistical dimensions that are independent of musical genre. The process of blending genres or other control dimensions is controlled in real-time by the user, facilitating intuitive exploration of a very high-dimensional state space.

In this paper, we make the following four contributions:

1) We describe two approaches for extracting axes of variation from musical genre databases: one based on maximizing the distance among musical models, and one based on principal components analysis.

2) We introduce a novel user interface component (the polygon slider) for controlling several related parameters, and we demonstrate its application to our learned parameterization of musical chord sequences.

3) We discuss the results of a user study supporting the hypothesis that an automatic approach to defining control axes provides a better user experience than the direct use of genre labels to define control axes.

4) We make our learned statistics available to other researchers in an online repository.

While we present this work in the context of musical chord sequences, we expect that the user interaction paradigm developed here has broad applicability in domains requiring user-guided creative exploration.

RELATED WORK

Creative domains are by nature open to a wide range of expression: as a result, there are typically many degrees of freedom that can be manipulated in the creative process. This limitless variety can be difficult even for professional artists to manage, and as such there have been efforts in various domains to reduce the complexity to allow for easier control. Some of these efforts have been manual, as in sound synthesis, where early systems requiring moving patch cables, and hundreds of parameters were often reduced to only the most commonly used [4]. More recently, there have been computational efforts to make simplifications in a variety of domains. For instance, in computer animation, there are hundreds to thousands of motion controls for a complex figure; recent work has attempted to simplify this space. For example, Liu et al. [7] present a method for learning "style" parameters from motion capture data, which can then be used to control the nature of characters' motion paths.

Looking specifically at the space of music composition, there have been several efforts to parameterize musical style to allow for easier exploration of the range of musical creations. One natural set of controls has is that of emotion, and there has been a variety of work on mapping emotional parameters for music synthesis. For instance, Wallis et al. [13] create an explicit mapping between emotional parameters (arousal and valence) and musical attributes; they can then choose or move between arbitrary points in this space to generate new music appropriate to this setting. Legaspi et al. [6] first learn a perceptual model of how each user maps musical features to emotional states; users can then specify emotional states and have the system generate appropriate music.

Our approach is more data-driven: we attempt to learn a parameterization of styles from existing pieces of music. Furthermore, we are restricting our notion of style to a model of the chord transitions therein. This notion has been developed in a variety of past research: Simon et al. [11] present an automatic accompaniment system that learns chord transition models from a small database of songs and allows for a heuristically-assigned axis of style variation (a major/minor factor). Chuan and Chew [2] present a system for learning a style from a small number of examples by using a combination of transition statistics and musical knowledge; they then use this model to generate new accompaniments. Allan and Williams [1] make use of a chord model learned from Bach chorales to provide automatic harmonization of melodies in the style of Bach. Cope [3] uses music theory and Augmented Transition Networks to compose music in the style of pieces in a hand-selected source database. Cope also demonstrates a provocative example of database-blending in the piece "Mozart in Bali", which uses data-driven methods to generate not only melody and harmony, but entire works.

SYSTEM AND USER INTERFACE

The goal of the present work is to enable rapid exploration of musical chord sequences using a high-level abstraction of this large and complex space. We achieve that goal by extracting meaningful axes of variation from a database of music; those axes are controlled by a user to create new chord sequences to accompany a vocal melody.

In subsequent sections, we will explore two methods for extracting these axes of variation from a database of music. Before we introduce these methods, however, we will describe the software system and user interface that are used to explore chord sequences for *any* set of control axes. The system and user interface described here are used throughout our evaluation.

Chord Generation

We adapt the method of Simon et al. [11] as our basic framework for generating a chord sequence to accompany a vocal melody. This method uses a first-order Hidden Markov Model (HMM) to model a chord sequence, where nodes in the model are chords in the sequence, and the model's observations are fragments of a vocal melody. Given observations (a sung melody), this method uses the Viterbi algorithm to select optimal chords for accompanying that melody. This method was selected as our starting point (over related methods by Allan and Williams [1], Chuan and Chew [2], and others) for its straightforward use of a Markov transition matrix as its core data representation, which – as we will discuss shortly – offers a natural mechanism for describing control axes. This method has also been demonstrated to work with vocal melodies, allowing us to build an experimental environment that simulates a songwriting experience better than a system which depends on symbolic melodies.

We compute observation probabilities (the relationship between chords and melody) according to Simon et al. However, we introduce novel mechanisms for determining and manipulating *transition probabilities among chords* using high-level control axes that are learned from data. The remainder of this paper will assume that our goal is to derive these control axes, construct Markov transition matrices (describing chord transition rules) from them, and intuitively present these axes to a user. For a more detailed description of the complete algorithm for generating chords given a transition matrix and a melody, see [11].

The Polygon Slider

Given a model parameterized by several continuous-valued control axes, a user manipulates these control axes through the interface component shown in Figure 1, which we refer to as a *polygon slider*. If we have *N* parameters, we draw *N* axes emanating from a central circle, spaced equally around the circle. Each axis functions as a traditional slider control: the small "handle" on each axis can be dragged along the axis from the inner circle to the larger outer circle. We use the position of the handle along the axis to determine the value of the associated parameter. The slider positions on neighboring axes are connected to form a polygon, and the interior of that polygon is filled with color to give a concise visual representation of the current settings.

Parameter manipulation could have been implemented as a series of linear sliders, but the polygon slider offers several benefits over linear sliders:

1) Rapid editing: our goal is to enable rapid exploration of the chord sequence space; placing the control axes in a single compact widget allows rapid manipulation of the available parameters with minimal mouse movement.

2) Dynamic layout: this widget adapts its layout dynamically to varying numbers of axes, without changing its required screen space. This is important in a scenario where a user dynamically adds clusters based on available data or personal preference.

3) Integrated visualization: Parameter *editing* is tightly linked to a descriptive *visualization* of the current parameter space (the area and shape of the polygon).

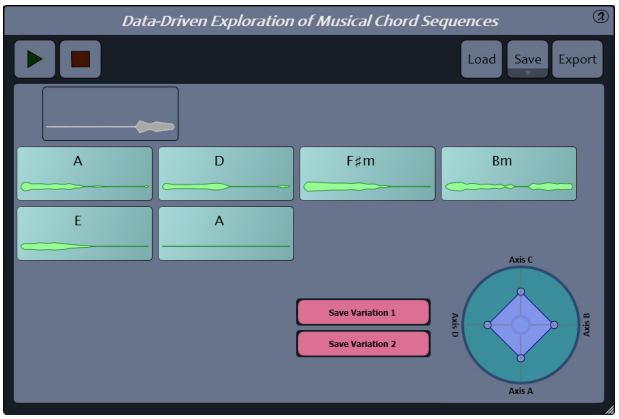


Figure 2: The user interface presented to participants during our evaluation. Participants could see the current chord sequence and used the polygon-slider (lower-right) to explore chord sequence variations. During our evaluation, the control axes were given "anonymized" labels such as "Axis A".

System Integration

The chord generation procedure and the polygon slider for manipulating control axes are integrated into the system shown in Figure 2. A vocal melody is recorded or loaded from a file, and accompanying chords are selected and shown to the user in a central display area. Manipulating the polygon control, which is shown in the lower-right, changes the chord sequence in real-time. Users can listen to the voice along with the current chord sequence, which is rendered as an instrumental accompaniment using a commercial accompaniment engine [9]. The two "save variation" buttons allow the user to save two chord sequences for the current melody; this feature is used in our evaluation, which is described below.

ASSIGNING CONTROL AXES

In this section, we describe two approaches to implementing the control axes (i.e., defining transition matrices based on the control axes) for the system described in the previous section. One approach is based on model-driven clustering, the other on principal components analysis. Before presenting these novel approaches, however, we will discuss a simpler scheme: using manually-assigned genre labels to build models for different genres. The drawbacks of this simple scheme will motivate our novel solutions.

Manually-Assigned Genre Labels

An obvious method for building statistical models that can enable user-driven exploration is to use hand-assigned genre labels. At a high level, one might build a statistical model for "country" music, a statistical model for "rock"

music, etc., then assign each of these models to a control axis and blend the models according to the axis values.

More formally, given a set of N genre-specific databases, we compute the transition probability matrix for each database by counting all chord transitions in that database[1] (as per Simon et al.). Then each element in our transition matrix is a linear combination of the corresponding elements in the genre-specific transition matrices:

$$P(c_i \rightarrow c_j) = \sum_d w_d P_d(c_i \rightarrow c_j), \text{ where } \sum_d w_d = 1 \quad (1)$$

Here $P(c_i \rightarrow c_j)$ is the probability of chord j following chord i in a sequence (for example, the probability of a "C major" chord following an "E minor" chord), d is the set of genre-specific databases, and w_d is the user-defined weight for database d (i.e. the value from 0 to 1 of the corresponding polygon slider axis). Weights are normalized such that they sum to one, to ensure that manipulating the control always maintains a valid Markov transition matrix.

This would ideally provide a user experience like that shown in Figure 1a, for instance, where a user can blend among five different genres. However, there are two major drawbacks to this approach:

1) Most importantly, while chord transition statistics differ among genres, there is also a great deal of commonality among genres, corresponding to the traditional patterns of musical harmony. Therefore, if we prepare our control axes according to this scheme, all axes *primarily supply the same statistics*, making it difficult to explore a variety of chord sequences. Pilot experiments confirmed this: users were able to create very little variation among chord sequences by manipulating solely the genre-based axes. This motivates an approach that emphasizes the *differences* among genres more than the *similarities*.

2) Furthermore, this approach also requires labeling of the entire database, which is potentially tedious and is itself subjective and unreliable. This motivates a fully-automated approach.

Automatic Clustering

We would like to preserve the intuitive nature of genres as axis labels yet still create enough variation among axes to allow for interesting exploration (i.e., solve drawback (1) above). We therefore propose an approach that takes the initial clusters of songs labeled by genre (as used in the previous section) but allows songs to move among the clusters to maximize the variability among the derived control axes. This results in control axes that emphasize the most prominent characteristics of each initial cluster.

[1] Throughout this paper, all songs are transposed to the key of C before analysis, so all analyses are independent of key.

Distance metric

In order to maximize the variability among control axes, our clustering procedure maximizes the distances among clusters according to a metric directly tied to the transition matrices derived from those clusters. Specifically, the metric we use to compute the distance between two clusters is the sum of absolute differences between the normalized matrices of chord transition counts:

$$\text{AbsDiff}(A,B) = \sum_i |A(i) - B(i)|, \text{ where } \sum_i A(i) = 1 \text{ and } \sum_i B(i) = 1 \quad (2)$$

Here A and B are the matrices of chord transition counts corresponding to each cluster. That is, each row in A corresponds to a chord c_i, each column in A corresponds to a chord c_j, and the value of A(i,j) is the total number of times c_i is followed by c_j in all the songs in this cluster. A is then normalized so the sum of *all* elements is 1. Note that this is not the same as the *transition matrix* derived from A, where individual *rows* sum to 1. In other words, we use the *joint* probabilities of chord transitions for clustering, rather than the *conditional* probabilities that are actually used to generate new chords. This is important, as the joint matrices take into account the relative frequency of a given transition with respect to other transitions in other rows, whereas in the conditional matrices each row is normalized, so this information is lost.

We also experimented with a metric that maximized KL divergence among normalized count matrices, rather than the absolute difference, but the results were empirically very similar and slightly slower to compute, so we use the AbsDiff metric shown in Equation 2 for our evaluation.

Cluster seeding

We use genre labels to define initial clusters; i.e. songs labeled as "rock" are assigned to one cluster, songs labeled "country" to another cluster, etc. For the remainder of this paper, we will refer to clustering using the AbsDiff metric and this seeding approach as "Genre+AbsDiff" clustering.

We highlight that this approach fundamentally still depends on genre labels to initialize clusters, and therefore does not solve drawback (2) above. I.e., this approach is only *semi-automatic*. We can also begin with *random* clusters instead, and we hypothesized that this would lead to interesting control axes as well, though axes derived from random seeds can no longer be described with intuitive, genre-based labels. In order to assess the potential for a fully-automated approach to defining control axes, we also include this "Random+AbsDiff" clustering approach in our evaluation.

Clustering Algorithm: Summary

We summarize our clustering procedure in pseudocode as follows. Note that throughout this description, we will refer to "computing the distance between two clusters". We write this to be concise, but this refers more precisely to

"computing the distance between the transition matrices derived from two clusters", as per Equation 2.

Start with a collection of songs where each is assigned to one of *N* clusters. This initialization may use genre labeling (for Genre+AbsDiff clustering) or random partitioning (for Random+AbsDiff clustering).

For each cluster C_1

Compute the total distance between *all* pairs of clusters, according to the AbsDiff metric (Equation 2). This is our "baseline axis variability" V_{base}.

For each song S_1 in cluster C_1

For each *other* cluster C_2 != C_1

Temporarily re-assign S_1 from C_1 to C_2

Re-compute the total distance between *all* pairs of clusters, according to the AbsDiff metric (Equation 2). This is our "modified axis variability" $V_{mod}(C_2)$.

Assign song S_1 to the cluster that maximizes variability (V_{mod}). If all V_{mod} values are less than V_{base}, leave this song in cluster C_1.

Iterate until clusters are stationary (i.e., V_{base} is always greater than V_{mod}) or the total change in V_{base} over a complete loop over all clusters is less than a minimum threshold.

The clusters generated from this variability-maximization procedure are treated in the same way as the genre-based clusters described in the previous section: each axis controls the blending weight of one cluster, and transition matrices derived from the clusters are blended as per Equation 1.

This clustering scheme is similar in spirit to previously-described clustering methods that attempt to maximize the minimum distance between two points in different clusters [5]. However, unlike typical clustering applications, we are not ultimately interested in the assignments of individual points (songs) to clusters. Rather, the property we are interested in – expressive variability of control axes – is a property of *all* points in a cluster (the chord transition matrix). Therefore, our clustering method is unique in that it maximizes a function of entire clusters, not individual points. Consequently, conventional algorithms for maximizing more standard metrics do not apply, motivating the greedy scheme described above.

Principal Components Analysis

Another *fully automated* approach to implementing the control axes uses principal components analysis (PCA), instead of using the blending equations of Equation 1.

To create PCA-based control axes, we compute a chord transition probability table (by counting all chord transitions) for each individual song in our entire database (independent of genre). If there are m chords in our dictionary, we "unroll" each $m \times m$ transition matrix into a vector of length m^2 so that each song is represented as a vector in m^2-dimensional space. Then we perform PCA on this set of vectors, and extract the top N principal components, where N is the desired number of control axes.

A new transition matrix can be generated by adding the mean transition matrix to a linear combination of the principal components. We map the weight of each component to an axis of the polygon slider. The axes are defined to represent values from $-\sigma_k$ to σ_k (instead of 0 to 1 as in the cluster-blending approach), where σ_k is the standard deviation for component k. Generating a transition matrix to use for chord generation is then again a simple linear blending of vectors in m^2-dimensional space:

$$V(i) = V_0(i) + \sum_{k=1}^{N} w_k V_k(i) \qquad (3)$$

Here i ranges from 1 to m^2, V is the resulting blended vector (length m^2), w_k is the user-defined weight for axis k, V_k is the k^{th} principal component derived from the database (length m^2), and V_0 is the mean transition matrix, unrolled into a vector (length m^2). For PCA we also remove the constraint that weights sum to 1.

We then compute a transition matrix P by simply performing the opposite of our "unrolling" procedure, i.e. we create an $m \times m$ transition matrix by taking elements column-wise from V, enforcing a small minimum probability ε, and finally normalizing rows to sum to 1.

EVALUATION: METHODOLOGY

We conducted a usability study to evaluate our approaches to defining control axes for creative exploration of chord sequences. We sought to evaluate the relative benefits of using genre labels alone to define control axes vs. using either our clustering-based or PCA-based approaches.

Demographics

10 participants volunteered for a one-hour experiment, and received a $10 gift coupon as compensation for their time. All participants indicated some familiarity with musical chord sequences; this was a recruiting criterion.

Experimental Procedure

Eight short, original melodies were recorded by the same singer prior to the study. For each melody, participants were asked to use the software to generate two chord sequences to accompany that melody which were subjectively appropriate for the melody but as different as possible from each other.

Participants selected accompaniments by manipulating the polygon slider in the software shown in Figure 2. Participants were given 5 minutes to use the software for each melody, and they could play and stop the song (melody and accompaniment) as many times as they chose

within that period. Two buttons were provided to save the two variations for each melody required for the task.

For each participant, we randomly divided the songs into four groups of two songs per group. For each pair of songs, the software was configured into one of four experimental conditions (see below). Experimental conditions differed only in the mechanism by which the polygon slider's control axes were defined.

Experimental Conditions

We used four different versions of the program, where axes controlled the relevant parameters for the following approaches to defining control axes:

1) Clusters based on manual genre labels ("Genre")
2) Genre seeding, AbsDiff clustering ("Genre+AbsDiff")
3) PCA
4) Random seeding, AbsDiff clustering ("Random+ AbsDiff")

We remind the reader that "Genre+AbsDiff" refers to seeding clusters using genre labels, then maximizing distances among clusters using the AbsDiff metric. "Random+AbsDiff" refers to performing the same procedure but with *random* initial clusters.

Axes were labeled "anonymously" in all conditions, i.e. participants did not how the polygon slider axes were derived, and axis names were simple "Axis A", "Axis B", etc. For consistency in evaluation, we used four axes in each condition. Control axes in all four conditions were built from the same database: a private collection of 2556 MIDI files, hand-labeled as one of four types (798 Pop, 1444 Rock, 97 Country, 217 Beatles). We highlight that although there is a significant difference among the sizes of our initial genre sets, none of our approaches demand consistency in number of songs in each database, nor would that be reasonable to expect from any real database of labeled music. We also note that although we have referred to "genre" throughout this paper for convenience, there is no need that initial labels correspond to "genre" in a traditional sense, nor is "genre" objectively defined. In order to explore a reasonably stylistic range of axis types, we included one artist label ("Beatles") in addition to our labels based purely on genre ("pop", "rock", "country").

Chord labels were extracted from MIDI sequences using the method of Sleator and Temperley [8].

Data Collection

After each condition (two songs) was completed, participants filled out a condition-specific questionnaire eliciting subjective responses to the tool they had just used. At the end of the session, they filled out an overall questionnaire comparing all four versions of the program. The software was instrumented to log all polygon slider manipulations and all chord sequences generated.

EVALUATION: RESULTS

Questionnaire Responses

Questionnaire responses confirm that participants felt they were able to produce good chord sequences using the polygon slider, and strongly preferred the axes generated with our semi-automated (Genre+AbsDiff) and fully-automated (Random+AbsDiff) approaches to those generated using genre labels alone. PCA-based axes were scored at an intermediate level of preference. This is illustrated by responses to both Likert-scale questions and explicit preference ranking.

Likert-scale questions

After each of the four conditions, participants were presented with three statements about the exploratory power of the condition they had just completed:

- "I was able to flexibly explore chord sequences using this tool"
- "I was able to generate interesting variations in chord sequences using this tool"
- "I was able to produce good chord sequences using this tool".

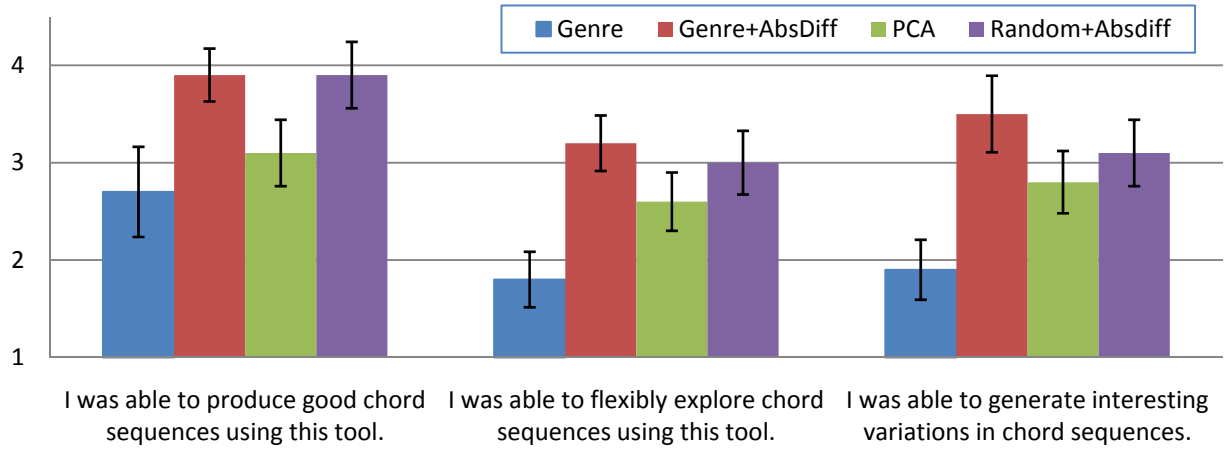


Figure 3: Mean responses to Likert-scale questions by experimental condition (1 = strongly disagree, 5 = strongly agree). The Genre+AbsDiff approach was preferred according to all three Likert-scale questions. Error bars indicate 95% confidence.

Participants were asked to respond to each question on a 5-point Likert scale (1 = strongly disagree, 5 = strongly agree). Mean responses are presented in Figure 3.

Differences among mean responses to Likert-scale questions were analyzed using t-tests at the 95% significance level, with correction for multiple comparisons using Holm's sequential Bonferroni procedure.

After correction for multiple comparisons, participants reported significantly higher agreement with all three statements after using the "Genre+AbsDiff" condition than the "Genre" condition. In other words, our AbsDiff procedure for maximizing variability among axes improved participants' subjective experience relative to the hand-assigned genre labels.

Agreement with the statement "I was able to flexibly explore chord sequences using this tool" was also significantly higher for the *randomly*-seeded clusters enhanced with our AbsDiff maximization procedure than for the hand-labeled genre clusters.

In addition to this *relative* analysis, we also specifically highlight the absolute values of the mean responses to the statement "I was able to produce good chord sequences with this tool" for the "Genre+AbsDiff" and "Random+AbsDiff" conditions: 3.9 in both cases. The median response was "agree" (4) for this question in both conditions. This establishes that participants felt they could produce good chord sequences with this system; i.e. that the underlying infrastructure was sufficient for the task. In fact, for the Genre+AbsDiff condition, no participants "disagreed" or "strongly disagreed" with this statement.

Preference ranking

At the end of the session, participants were asked to rate the four conditions – identified only as "A", "B", "C", and "D" – according to their overall preference. Mean rankings (where 1 is "most preferred") are shown in Figure 4.

Importantly, *no* participants chose the basic genre clusters as their most-preferred or even second-most-preferred condition, supporting our core hypothesis that traditional metadata is insufficient to allow compelling genre-specific exploration. The variability-maximized genre clusters (Genre+AbsDiff) were the most popular choice as "most preferred".

Differences among preference rankings were analyzed using the Kruskal-Wallis nonparametric one-way ANOVA, which allowed us to reject the null hypothesis that preference rankings were random at a confidence level of $p < 0.0001$. Post-hoc testing with correction for multiple comparisons revealed significant differences in the ranking of the basic genre clusters and the rankings of the AbsDiff-based conditions (i.e., participants significantly preferred the clusters that had been subjected to our AbsDiff variability-maximizing procedure). Furthermore, the "Genre+AbsDiff" condition was also significantly preferred over the PCA condition.

Number of Sequences Explored

Another metric for comparing the four conditions (i.e., the four systems for defining control axes) is the *total number of unique sequences* explored during the session for each condition. We hypothesize that making more unique sequences available will maximize the system's potential for creative exploration. We therefore computed the number of total sequences explored for each condition. To avoid

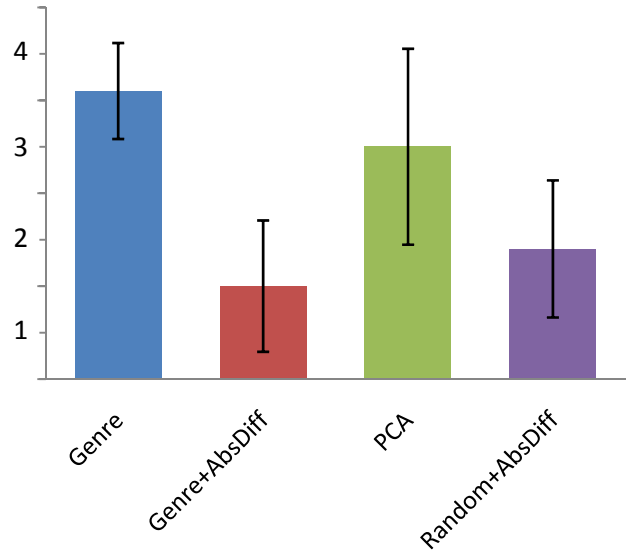


Figure 4: Mean preference ranking (*lower rank* is *more preferred*) for each experimental condition. The "Genre+AbsDiff" approach to generating control axes was the most preferred by participants.

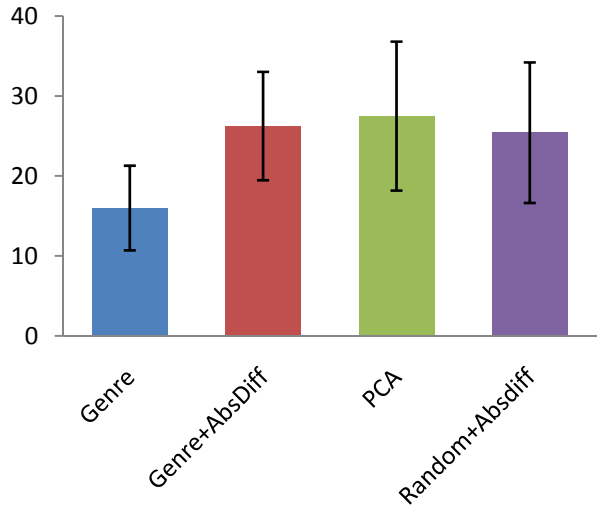


Figure 5: Mean numbers of unique sequences explored in each condition. Participants explored fewer sequences using the purely-genre-based transition matrices than using the matrices derived from our automated or semi-automated approaches.

transient states that were not considered by the participant, we did not count sequences that were displayed on-screen for less than three seconds. Results are shown in Figure 5.

Differences among numbers of unique sequences explored were analyzed using t-tests at the 95% significance level, with correction for multiple comparisons using Holm's sequential Bonferroni procedure.

After correction for multiple comparisons, participants explored significantly more unique sequences with all automated or semi-automated methods ("Genre", "Genre+AbsDiff", and PCA) than with the genre-based clusters alone (all $p < 0.01$), confirming our hypothesis that using genre-based clusters alone does not capture enough musical variation for interesting exploration.

VISUALIZING CHORD TRANSITIONS

To give the reader a better sense of the contents of our chord transition matrices and the effect of our clustering procedure, we present Figures 6 and 7, which depict transition matrices for two genres ("pop" and "rock") before and after AbsDiff clustering.

Figure 6 shows the transition matrices used in the "Genre" condition. While some differences are visible (such as the higher-probability E→Bm transition in the "rock" matrix), the matrices show a high degree of similarity.

In contrast, Figure 7 displays the matrices computed after "Genre+AbsDiff" clustering. Prominent vertical bands in Figure 7 indicate that certain chords in each matrix – chords that are characteristic of the original genre clusters – are now more likely to be the destination of a transition. For example, in the "pop" case, chords characteristic of the minor mode are prevalent, including Am, Dm, Em, and E. Some study participants (see below) noted that this control axis affected the minor feel of the output.

DISCUSSION

Summary of Results

Our results show that providing a series of automatically-defined control axes provides a useful mechanism for exploring the high-dimensional space of chord sequences, and that even four axes was enough to allow users a broad exploration of this space.

While one might expect that defining control axes based only on genre labels would give an intuitive mechanism for exploration, our results show that a more sophisticated approach is required to create axes that have interesting exploratory behaviors: we specifically show that using a clustering mechanism to emphasize differences among genres produces a more powerful exploratory tool. In fact, even a clustering procedure that begins with *random* seeds yields a more powerful exploratory tool than axes based purely on genre, according to participant responses.

Inferring Axis Definitions

It is interesting to ask whether participants were able to form intuitions for what the axes meant (we remind the reader that in all conditions, the axes were labeled with "anonymous" strings: "Axis A", etc.). To address this question, we solicited additional data in each of the four condition-specific questionnaires. For each condition, participants were asked: "Please assign 1-2 word names to each axis on the control". Responses were free-text.

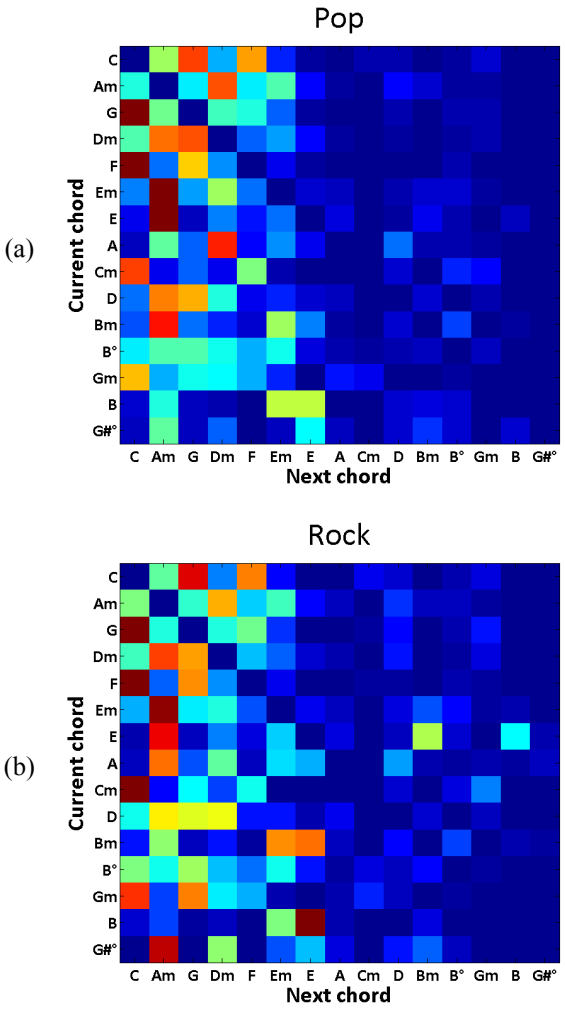


Figure 6: Example transition matrices, before AbsDiff clustering: (a) pop, (b) rock. These matrices correspond to the "genre" condition; i.e. they represent transition probabilities extracted from a hand-labeled database, without AbsDiff clustering. Red and blue indicate high- and low-probability transitions, respectively. Rows are normalized (a transition matrix uses conditional, rather than joint, probabilities), so color does not reflect the overall probability of an individual chord, but rows and columns are sorted by overall chord frequency in the database, i.e. "C" was the most common chord, followed by "A Minor", etc. Only the 15 most common chords are shown here. All songs are transposed to the key of "C" before analysis, so this figure is independent of key. Diagonals (self-transitions) are not well-defined in chord sequences, so they are set to zero for this visualization. This figure is presented to contrast with the post-clustering results in Figure 7, and to summarize the data available in our online repository.

The "Genre" condition (using genre labels directly) not only failed to elicit genre-specific responses, but generally failed to elicit responses at all: i.e., the most common response in this condition was either to leave the question blank or a null response such as "They all pretty much do nothing". However, the other conditions – based on clustering or PCA – resulted in some common responses, indicating that participants developed their own intuitions for how to navigate chord transition space using the control. Some examples responses highlight the type of intuitions respondents developed, which ranged from advanced music-theoretic concepts (such as "Dorian mode") to very abstract descriptions (such as "dark"):

Genre:

- Axis A (Pop): "more of an ending sound", "relative minor/major", "minor"
- Axis B (Beatles): "1st repetition sound – expected more after"
- Axis C (Rock): "weirdness", "more of an ending sound", "variation/consistency"
- Axis D (Country): "1st repetition sound – expected more after", "major"

Genre + AbsDiff:

- Axis A (Pop): "blandness", "relative minor", "influences the minor feel of the song"
- Axis B (Beatles): "# of ii→IV's", "Dorian mode", "tension"
- Axis C (Rock): "minor", "adds AABA structure"
- Axis D (Country): "changed last chord", "adds some minor", "atonality (chords would be C→Cm)"

PCA:

- Axis A (PC1): "mood progression", "changes variation", "variation/repetition"
- Axis B (PC2): "more minors", "determine which note to harmonize within the measure"
- Axis C (PC3): "complexity", "mood"
- Axis D (PC4): "minor"

Random + AbsDiff:

- Axis A: "complexity", "dark", "major vs. minor", "sharp minor chords", "changes key", minor chords
- Axis B: "changes chord sequence", "traditional", "mix of major/minor"
- Axis C: "minor", "sharp minor chords", "major vs. minor", "traditional/happy", "major chords"
- Axis D: "amount of variation in chords", "not-in-key chords", "weird", "modulate"

The breadth of these responses demonstrates that given abstract axes for creative exploration, users *do* build intuitions for those axes, even when the axes are defined by fairly abstract mechanisms such as PCA.

The most common theme of responses in all conditions involved the notion of major and minor chords. 18 of the responses conjectured that an axis controlled something related to the distribution of major vs. minor chords in the output. This may indicate that these are such strong preconceptions of the dimensionality of chord sequences that participants specifically sought to assign at least one axis to "major-ness", or it may indicate that these concepts are in fact identified by the clustering procedure and extracted as meaningful axes. The present work is primarily concerned with the process of defining and evaluating effective control axes, but future work will explore the statistical basis of these axes and their relationship to music-theoretic constructs like "major" and "minor".

Observations of participants during the study indicated some frustration with the "PCA" condition. Many participants felt that the sliders were "too sensitive' in this case; deviations from the mean easily seemed to lead to

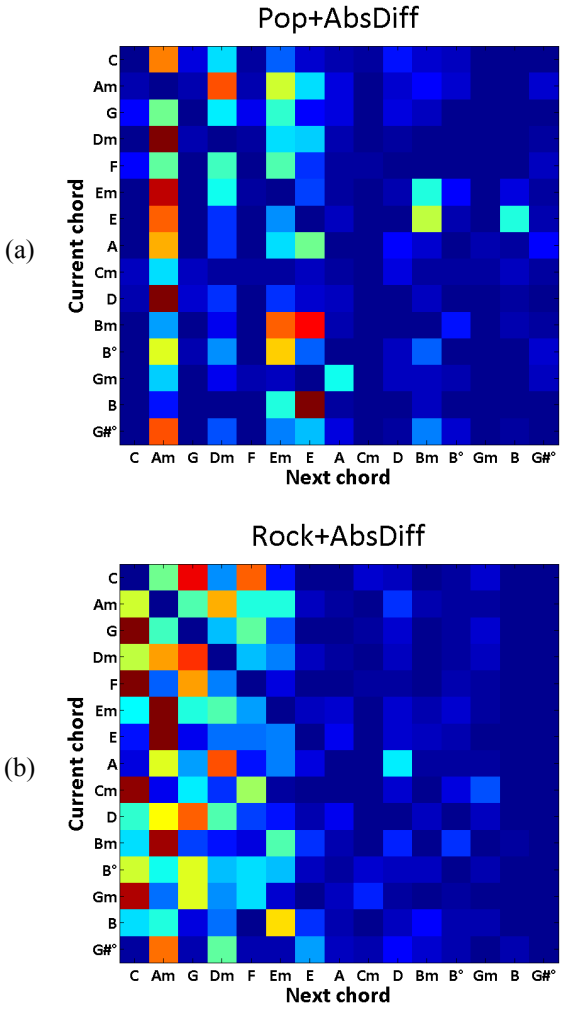


Figure 7: Example transition matrices, after AbsDiff clustering: (a) pop, (b) rock. These matrices correspond to the "Genre+AbsDiff" condition in our experiment. See Figure 6 for a detailed description of the figure coloring and layout. In comparison to Figure 6, we see that AbsDiff clustering has made the two matrices more different, and – particularly in the pop matrix – has enhanced specific characteristic chords, manifested here as vertical banding.

unpredictable parts of transition matrix space, suggesting that clustering leads to more intuitive axes.

Free responses

When asked for additional comments or suggestions in free-response questions throughout the survey, may participants indicated the desire to identify which chord to change or which chords to hold fixed while varying the parameters; e.g. "I'd like to be able to lock in chords for most of the song, but just tweak a few measures", or "I'd like to be able to control phrases more independently". Other interesting responses included:

- "I like the variety of progressions that came out of condition 2 [Genre+AbsDiff]."
- "The axes in condition 3 [PCA] seemed especially sensitive, and also very different in action from the other variations."

When asked the free-response question "Would the ability to explore chord sequences with a control similar to this one be useful to you?" participants were generally positive:

- "Yes, possibly… I'd be interested in something [that could] suggest tonal possibilities I hadn't thought of."
- "Yes. It's too easy to fall into patterns and predictability."
- "Yes, but felt limited in that all changes seemed to affect entire song."
- "Yes… I can see this being very helpful for filling out chords for a jazz tune."
- "Definitely."
- "Possibly—I'd like control over key and frequency of change (not just one chord per measure)"
- Yes. Gives interesting new ideas for chord progressions."

Some participants were frustrated, however, at not knowing what the axes actually did, due to the blind study.

DATA REPOSITORY

In order to facilitate future systems leveraging these analyses, future musicological analysis of chords in popular music, and replication of our results, we have made our transition matrices for all four experimental conditions available at:

http://research.microsoft.com/users/dan/chords

FUTURE WORK

Our fully-automatic approaches to axis definition (PCA, Random+AbsDiff) do not require manual labeling of training data, but also do not naturally allow end-user customization. Our semi-automatic approaches, however, generalize naturally to a scenario where an end-user directs the axis-definition process itself as part of the creative process. Future work might include, for example, allowing users to define axes by specifying examples that induce specific emotions, examples from their favorite artists, etc.

We are also quite interested in applying this work to creative exploration of *melodies*, in addition to *chord sequences*. Previous work on automatic melody generation has also leveraged Hidden Markov Models, so we hope to explore applying of our approaches to defining HMM parameters for melody generation.

We are also very interested in applying the principle of parameterized exploration to other creative disciplines. Future work will include engaging experts from other creative domains – such as painting and creative writing – to more systematically identify exploratory processes that might be appropriate for parameterization.

ACKNOWLEDGMENTS

We thank Yuval Peres, Eyal Lubetzky, Paul Cuff, and David Wilson for discussions on sequence modeling.

REFERENCES

1. Allan, M., Williams, C.K.I. Harmonising Chorales by Probabilistic Inference. *Proc NIPS 2005*.
2. Chuan, C.-H., Chew, E. A Hybrid System for Automatic Generation of Style-Specific Accompaniment. *4th Intl Joint Workshop on Computational Creativity*, June 2007.
3. Cope, D. *Computers and Musical Style*. A-R Editions: Madison, WI, 1991.
4. Dodge, C., Jerse, T.. *Computer Music: Synthesis, Composition, and Performance*. Schirmer, 1997.
5. Jain, A. K., Murty, M.N., and P.J. Flynn. Data Clustering: A Review. *ACM Computing Surveys* 31(3). September, 1999.
6. Legaspi, R., Hashimoto, Y., Moriyama, K., Kurihara, S., Numao, M. Music compositional intelligence with an affective flavor. *Proc IUI 2007*.
7. Liu, C. K., Hertzmann, A., and Z. Popović. Learning Physics-Based Motion Style with Nonlinear Inverse Optimization. *ACM SIGGRAPH 2005*.
8. Morris, D., Simon, I., and Basu, S. Exposing Parameters of a Trained Dynamic Model for Interactive Music Creation. *Proc AAAI 2008.*
9. PG Music Inc: Band-in-a-Box. http://pgmusic.com
10. Rohrmeier, M., and Cross, I.. Statistical Properties of Tonal Harmony in Bach's Chorales. *Proc 10th Intl Conf on Music Perception and Cognition* (ICMPC10). Sapporo, Japan, 2008.
11. Simon, I., Morris, D., and Basu, S. MySong: Automatic Accompaniment Generation for Vocal Melodies. *Proc ACM CHI 2008*.
12. Sleator, D. and Temperley, D. 2001. The Melisma Music Analyzer. http://www.link.cs.cmu.edu/melisma/
13. Wallis, I., Ingalls, T., and E. Campana. "Computer Generating Emotional Music: The Design of an Affective Music Algorithm." *Proc Digital Audio Effects* (DAFx-08). Espoo, Finland. 2008.