Comparative Analysis of Key Inference Models for Musical Metacreation

Anonymized

Abstract

Creative musical systems must be equipped with certain intelligent abilities to understand fundamental aspects of music, particularly in order to autonomously interact with other creative agents. Key inference, a relatively simple task for trained human experts, is one such intelligent ability required to normalize and analyze melodic compositions. We assess the accuracy of several traditional and machine learning approaches to the key and key signature inference problems on a dataset of 480 melodies in MIDI format. We compare these accuracies with those of trained human musicians on the same tasks. We evaluate the impact of including note duration and note repetition as learning features. We find machine learning approaches outperform traditional key inference methods. The highest accuracies (0.729 for key inference and 0.896 for key signature inference) was achieved using a 4-gram language model. Including note duration improved the results of traditional approaches when inferring key, but had the opposite effect for key signature inference. Our findings suggest that the key of a melodic passage depends more heavily on the sequence of the notes rather than their frequency or distribution.

Introduction

A fundamental element of musical metacreation is the quest to "[endow] machines with creative behavior" (Pasquier, Eigenfeldt, and Bown 2012). Such behavior in human draws on a full spectrum of intelligent abilities (Colton, Wiggins, and others 2012). Often there are overlaps in the intelligent abilities that are required for different tasks. This has been noted to be especially true of musical computational creative systems (Bodily and Ventura 2017).

An example of such an ability is *musical key inference*: the ability to recognize the group of pitches or scale that form the basis of a particular composition. This ability is a relatively simple task for trained human musicians. The key or tonic of a passage is used to normalize passages, motifs, and progressions into a common language.

In computational systems data often comes pre-labeled with the key. However, humans and music systems alike are often placed in unsupervised training scenarios in which the key is not labeled, but must rather be inferred. This inference is critical, particularly for interactive systems which must interpret and respond to other music systems. The key in which systems improvise or compose is often not explicitly communicated by other performing agents but is rather inferred from what *is* explicitly communicated (i.e., notes, chords, etc.). In the quest for creating autonomous, interactive musical agents, to what extent can this skill of inferring musical key be learned?

The *key* or tonic is a root pitch and modality (major or minor) which forms the structural basis for tonal music. This tonality provides a context within which "the melodic and harmonic unfolding of a composition takes place" (Vos and Van Geenen 1996). Even the untrained ear appreciates the structure, dissonance, and resolution that tonality provides. Each key is associated with a *key signature* representing the specific pitches that belong to the scale represented by the key. There exists a two-to-one relationship between keys and key signatures with each major key and its *relative natural minor key* (whose root is three half steps lower) being associated with the same key signature. Though songs (less frequently) exist with keys whose modes are not strictly major or minor, we have for our purposes chosen to focus solely on these two more common modes.

Western pop music in particular presents an interesting case study because pop songs regularly and deliberately break rules of traditional modulation and are often found to end on chords that are either unresolved or are entirely unrelated to the key in which the rest of the song is written.

There is a profound irony in the contrast between knowing how and being able to explain how to infer a song's key. Expert musicians routinely and accurately infer a song's key. However the best description for their methodology often goes no further than to find the key that "feels" right, provides a sense of "finish", or the key where the song "lands".

In this context it has commonly been supposed that one need simply find the key which minimizes the total number of accidentals in a song. Part of our purpose is to test this hypothesis by comparing this approach with several machine learning algorithms.

Harmony plays an integral role in determining a song's key. However, in cases of interactive monophonic systems, even harmonic data may be unavailable. We hypothesize that

a monophonic melodic sequence alone is sufficient in most cases to determine a song's key. We test this hypothesis on a dataset of 480 pop melodies in MIDI format. As a fundamental building block of music knowledge and creativity, effective key inference models stand to add increase autonomy in musical metacreation systems.

Related Work

Several previous studies have examined key inference in various contexts, though to our knowledge our is the first attempt to do so using solely melodies and in the pop music domain.

Krumhansl matches the relative frequencies and durations with which tones are sounded (which she terms a *tonal hierarchy*) of a song against the known tonal hierarchies of each key (Krumhansl 2001). This algorithm was applied to infer keys for compositions from three classical composers, Bach, Chopin, and Shostakovich.

The key-finding algorithm of Longuet-Higgins and Steedman successively eliminates keys based on the presence or absence of the song's notes in each of the major and minor scales (Longuet-Higgins and Steedman 1971). Holtzman (1977) infers key from the prevalence of common key-defining intervals (e.g., triads, tonic-fifths, tonic-thirds). Both algorithms applied their algorithm to Bach's *Well-Tempered Clavier*.

Hu and Saul take an LDA approach to key-finding, looking for common co-occurrences of notes in songs. Their model essentially treats keys like topics. They then model songs as a random mixture of key-profiles, allowing them to track modulations (Hu and Saul 2009). Temperley interprets the traditional key-profile model as a Bayesian probabilistic model and discusses the implications of the connection between these two models (Temperley 2002). All applications of the model are on the Kostka-Payne corpus, a collection of textbook excerpts of tonal music. Vos and Van Geenen present a parallel search key-finding algorithm for single-voiced music. A song's notes are evaluated against both the scalar and the chordal structures of each key. They demonstrate the model's effectiveness on Bach's Well-Tempered Clavier (Vos and Van Geenen 1996). Zhu et al. present a method for key estimation in acoustic pop and classical music (Zhu, Kankanhalli, and Gao 2005). Their method performs marginally higher with pop music than classical music.

Much recent work has been devoted to inferring key from acoustic music. Shenoy et al. outline a rule-based algorithm for finding key from acoustic musical signals using chroma based frequency analysis and chord progression patterns (Shenoy, Mohapatra, and Wang 2004). Mauch and Dixon infer chords and key simultaneously from audio (Mauch and Dixon 2010). Chafe et al. extract symbolic music from audio from which meter and key are inferred (in that order) (Chafe, Mont-Reynaud, and Rush 1982). The key-recognizer assumes that rhythmic and melodic accent points are significant features for inferring key.

Several methods have been presented for identifying key signature. They have been used mostly in Classical music. Pop music uses different modulations and is unique in that often the song finishes on a chord other than the key the song is in. No studies were found to examine key inference in pop melodies.

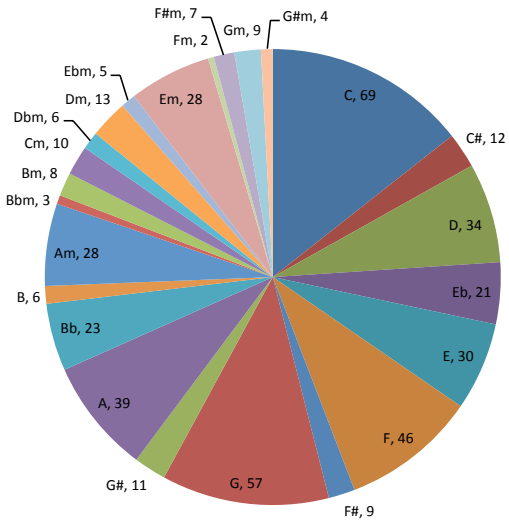

Figure 1: *Key distribution of pop melodies*. Every key is represented at least twice in the subset of melodies. In general keys with fewer sharps and flats are more highly represented.

Methods

Data

We collected 480 melodies from 278 pop artists representing music spanning several decades (see Figure 1). Songs were compiled from several online MIDI databases and keys (inferred from melody together with harmonic context) were manually labeled by trained musicians. We were pleased to find that every key was represented by at least two songs (see Figure 1). Only songs with a single key were selected for our experiments. Melody notes were isolated from the MIDI files.

Implementation

We compared the accuracy of 4 traditional and 5 machine learning methods against the accuracy of trained musicians on the tasks of inferring key and key signature for pop melodies. Each key signature there is a major and minor key and thus key inference—which precludes key signature inference—represents the more difficult of the two tasks. For the machine learning methods, we used 10-fold cross validation for training and testing.

Minimize Accidentals By Count. This algorithm represents the common theory that the best key is that which minimizes the number of resulting accidentals.

Minimize Accidentals By Duration. Similar to the previous method but finds the key which minimizes the total duration of accidentals.

RMSE of Pitch Count Profiles. Similar to the method followed by (Krumhansl 2001) and others, we generated pitch

Table 1: Highly Represented Pop Artists

Artist	# of songs
beatles	28
elvis presley	13
kiss	11
madonna	8
eagles	6
aerosmith	6
elton john	6
u2	6
beach boys	5
michael jackson	5
pink floyd	5
bobby vee	5
adele	5
queen	4
kinks	4

profiles. Rather than generate profiles for each major and minor key, we chose to generate a single profile for all major keys and a second for all minor keys. This decision is based on the assumption that the variation in pitch distribution varies very little (if at all) as compared to the distributional variation between the major and minor modes. Major and Minor mode pitch profiles were generated from the pitch counts in training instances normalized to either the C major or A minor keys depending on the whether the instance was major or minor. A pitch profile is created for each test instance, transposed into each of the 12 possible keys. Each transposed profile is compared to both the major and minor generic pitch profiles using root mean squared error (RMSE). The transposed pitch profile and the major/minor pitch profile with the minimum RMSE value are used to infer key and modality.

RMSE of Pitch Duration Profiles. Similar to the previous method but pitch profiles are generated from pitch durations rather than mere counts (see Figure 2).

n-*gram Models*. An n-gram model calculates the probability of the next token given some context window of length n. These probabilities are learned from the sequence of notes in the training instances and then used to calculate the probability of note sequences in the test instances. We trained n-gram models for values of n from 1 to 5, using Laplace smoothing and a pseudocount alpha value of 1. For each value of n a single n-gram model was trained for melodies in major keys and another for melodies in minor keys. Probabilities were normalized across both models. Training instances were than transposed and scored by each trained model. The transposition and model which maximized the probability of the training instance determined the key and modality.

In addition to the methods described above, we also report accuracy from four other sources. First, the *baseline* accuracy represents the approach of always guessing the most common class. *MIDI Annotations* refers to the key signature that was originally given in the MIDI file (if any was provided). We report the accuracy of a third-party MIDI-

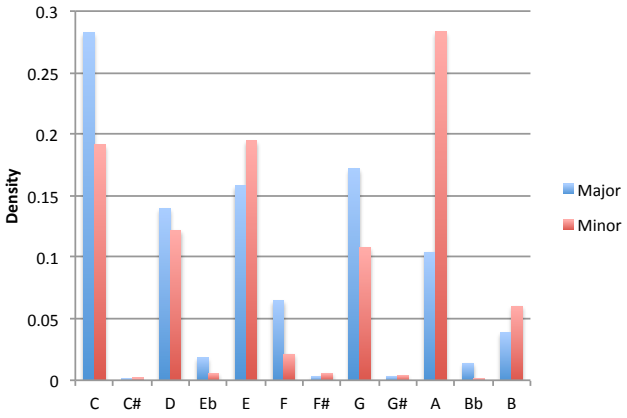

Figure 2: *Weighted Major/Minor Pitch Profiles*. Profiles are based on the duration of pitches in each of the major and minor modes. Profiles for major keys (blue) have been normalized to C major and those for minor keys (red) to A minor. Note that the A pitch is relatively more frequent for minor keys (where it functions as the tonic) than for major keys (where it functions as the 6th from the root). Likewise the G pitch is relatively more frequent for major keys (where it functions as the dominant) than for minor keys (where it functions as the 7th from the root).

reader called *MuseScore*. Insofar as MIDI is a format concerned primarily with generating audio, many files fail to include (accurate) information about key or time signature, thus motivating the need to infer this information from the notes themselves. This functionality is built in with varying success to many programs which render MIDI files as sheet music. MuseScore (version 2.0.3) is one such program and we include the accuracy of MuseScore's inferred key-signature in our results for comparison.

Lastly, we enlisted 15 trained musicians to complete the same melodic key inference task. Each musician was asked to listen to 20 melodies (1 of which was a simple control melody to assure quality survey data—all respondents answered the control correctly). In addition to labeling the key and key signature for each melody, respondents were asked to indicate how familiar they were with the melody. This was to control for bias that might arise as a result of prior knowledge of the harmonic context of the song. We report the accuracies on the melodic key inference task as a function of the degree of familiarity as well as an overall weighted average of these three scores.

Results

Results are shown in Table 2. We report accuracy both for inferring the key and for inferring the key signature. Inasmuch as key signature is a more generic classification of key (e.g., C major and A minor both have a key signature with no flats or sharps), accuracy for key signature will always be better than accuracy for key.

Table 2: Key Finding Accuracy for Pop Melodies

Method	Key	Key Signature
Baseline (C)	0.144	0.202
MIDI Annotations	n/a	0.483
MuseScore	n/a	0.746
Minimize Accidentals by Count	0.494	0.660
Minimize Accidentals by Duration	0.490	0.665
RMSE of Pitch Count Profiles	0.606	0.796
RMSE of Pitch Duration Profiles	0.613	0.765
Unigram model	0.629	0.815
Bigram model	0.654	0.883
Trigram model	0.679	0.883
4-gram model	*0.729*	*0.896*
5-gram model	0.700	0.885
Human ("I know this song")	0.846	0.865
Human ("Sounds familiar")	0.75	0.786
Human ("I don't know this song")	0.676	0.815
Human Average	0.719	0.822

Discussion

Normalizing to a common key really only requires that we identify the key signature for a composition without regard for whether the key is the major key associated with the key signature or its relative minor. We chose to model the major and minor separately based on the hypothesis that the melodies that each produces would be sufficiently different to warrant creating individual profiles.The key inference methods which minimize the frequency of accidentals show the most dramatic improvement because these methods inherently fail to provide a way of distinguishing between a major key and its relative minor.

We encountered several challenges unique to the pop music domain. Songs based on blues scales often include the flat seventh (which would suggest a key a fifth below the actual tonic) or both the major and minor third. Hard rock songs often exclude the third all-together, making it difficult to infer whether a major or minor key signature is more accurate. These confounding influences are reflected in the confusion matrices for classifications of songs in these genres (see Figure 3).

As regards the *n*-gram models, we note that the unigram model is essentially equivalent to a pitch profile rendered as an applied probability distribution, and thus it seems reasonable that it should perform about on par with the RMSE of Pitch Count Profile method.

We increased the value of *n* until we observed a decrease in accuracy. As is typical of *n*-gram models, as *n* increases, the model begins to essentially memorize more than can be generalized from the training data to the test data at which point the differences between instances become as significant as the differences between the key classes.

It is important to note that although our best accuracy on key-finding (.729) is significantly below the values reported by studies mentioned in the related works, the task of inferring key from melody is significantly more challenging than inferring key from songs which include harmony. This is evidenced by the relative accuracies of the algorithmic and human performances on the melodic key inference problem.

It should also be considered that human listeners that are familiar with a melody may more accurately infer key from having familiarity with the harmony also. This is evidenced in our results in the fact that the more familiar the human listeners were with the song being played, the higher the accuracy on the key inference problem. Notably the same was not necessarily true with the key signature inference, suggesting that without knowledge of the context, humans may confuse a major and relative minor for an unknown melody but still infer the correct key. This is also evidenced by the confusion matrix for the 4-gram model, shown in Figure 3. We therefore express confidence in the reported accuracies of these models.

It is of interest to note that the *n*-gram models generally outperformed the human models on the key signature inference task. The 4- and 5-gram models outperformed the human respondents on the key inference task when the human was otherwise unfamiliar with the song being played.

Our results suggest that considering pitch counts or durations is less effective than a model which considers the sequence of pitches. This agrees with our intuition insofar as many pop songs spend the bulk of their duration modulating through chords which are not the root and may not even be closely related to the root. Notions of resolution and finding where the song "lands" inherently suggest that the contour and progression of the notes matters more than their frequency. The key is often most clearly defined at the beginning and ends of musical phrases or the beginning and end of the song itself. Whereas the method of counting note frequencies fails to give higher weight to these defining regions of the melodic passage, this information is embedded within the probabilistic framework of *n*-gram models.

We find the superior accuracy of *n*-gram models to the MuseScore key-signature inference model to be particularly promising inasmuch as it suggests that an *n*-gram model might be used to improve the state of the art in industrial MIDI-reading software.

As regards key inference from monophonic pop melodies, we find that machine learning methods (*n*-gram language models in particular) perform better than traditional key-finding algorithms, though both improve upon baseline accuracy. In the future we hope to evaluate how well unbiased human listeners would perform on the same task. We also envision developing a framework for detecting key *changes* in pop melodies and for normalizing unlabeled melodic data for compositional analysis.

Conclusion

We have examined several solutions to the task of inferring key and key signature in monophonic music sequences. We found that for the particular case of monophonic sequences, the 4-gram model performed best at both identifying key and key signature. Endowing musical creative systems with intelligent abilities like key inference increases their autonomy. As key is a fundamental concept to many genres of

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
C	57	0	0	0	0	2	0	3	0	0	0	0	6	0	0	0	0	1	0	0	0	0	0	0
C#	0	9	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
D	0	0	26	0	0	0	0	2	0	1	0	0	0	0	5	0	0	0	0	0	0	0	0	0
D#	0	0	0	15	0	0	0	0	0	0	1	0	0	0	0	4	0	0	1	0	0	0	0	0
E	0	0	0	0	20	0	0	1	0	0	0	1	0	0	1	0	2	0	0	5	0	0	0	0
F	0	0	0	1	0	40	0	0	0	0	1	0	0	0	0	0	0	1	0	0	3	0	0	0
F#	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
G	0	0	1	0	0	0	0	49	0	0	0	0	0	0	0	1	0	0	0	5	0	0	1	0
G#	0	0	0	0	0	0	0	0	10	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
A	1	0	3	0	1	1	0	0	0	27	0	0	1	0	1	0	0	0	0	1	0	3	0	0
A#	0	0	0	0	0	0	0	0	0	0	21	0	0	1	0	0	0	0	0	0	0	0	1	0
B	0	0	0	0	1	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0
C	11	0	0	0	0	0	0	2	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0
C#	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
D	0	0	5	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
D#	0	0	0	9	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
F	0	0	0	0	0	3	0	0	0	0	0	0	1	0	0	0	0	9	0	0	0	0	0	0
F#	0	1	0	0	0	0	3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
G	0	0	1	0	1	0	0	5	0	1	0	0	0	0	1	0	0	0	0	19	0	0	0	0
G#	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
A	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	5	0	0
A#	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	7	0
B	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	1

Figure 3: *Confusion Matrix for 4-gram model*. Much of the mis-classification confuses major and minor modes for the same key. In cases such as hard rock which use open fifths and exclude the third completely the mode (i.e., major or minor) is difficult to establish.

music, this computational subconcept model can be modularly reused to improve musical metacreation across several domains.

References

Bodily, P. M., and Ventura, D. 2017. HBPL: a framework for debating, developing, and reusing foundational models of musical metacreativity. In *Proceedings of the Fifth International Workshop on Musical Metacreation*.

Chafe, C.; Mont-Reynaud, B.; and Rush, L. 1982. Toward an intelligent editor of digital audio: Recognition of musical constructs. *Computer Music Journal* 6(1):30–41.

Colton, S.; Wiggins, G. A.; et al. 2012. Computational creativity: The final frontier? In *ECAI*, volume 12, 21–26.

Hu, D., and Saul, L. K. 2009. A probabilistic topic model for unsupervised learning of musical key-profiles. In *ISMIR*, 441–446. Citeseer.

Krumhansl, C. L. 2001. *Cognitive foundations of musical pitch*. Oxford University Press.

Longuet-Higgins, H. C., and Steedman, M. J. 1971. On interpreting bach. *Machine intelligence* 6:221–241.

Mauch, M., and Dixon, S. 2010. Simultaneous estimation of chords and musical context from audio. *IEEE Transactions on Audio, Speech, and Language Processing* 18(6):1280–1289.

Pasquier, P.; Eigenfeldt, A.; and Bown, O. 2012. Preface. In *Proceedings of the First International Workshop on Musical Metacreation*.

Shenoy, A.; Mohapatra, R.; and Wang, Y. 2004. Key determination of acoustic musical signals. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 3, 1771–1774. IEEE.

Temperley, D. 2002. A bayesian approach to key-finding. In *Music and artificial intelligence*. Springer. 195–206.

Vos, P. G., and Van Geenen, E. W. 1996. A parallel-processing key-finding model. *Music Perception: An Interdisciplinary Journal* 14(2):185–223.

Zhu, Y.; Kankanhalli, M. S.; and Gao, S. 2005. Music key detection for musical audio. In *11th International Multimedia Modelling Conference*, 30–37. IEEE.