Key recognition using a variety of Algorithms

Laurens van den Bercken (4587545), Christian Veenman (4495705), Simone van Veen (4605993), Albert ten Napel (4087798)

Krumhansl-Schmuckler Key-Finding Algorithm

This algorithm tries to assign the right major or minor key to a piece of music by matching tone durations to a major and minor profile, given below [1,2].

do	do#	re	re#	mi	fa	fa#	so	so#	la	la#	ti
6.35	2.23	3.48	2.33	4.38	4.09	2.52	5.19	2.39	3.66	2.29	2.88

Major profile

la	la#	ti	do	do#	re	re#	mi	fa	fa#	so	so#
6.33	2.68	3.52	5.38	2.60	3.53	2.54	4.75	3.98	2.69	3.34	3.17

Minor profile

This profiles come from experiments done by Carol L. Krumhansl and Mark A. Schmuckler. A set of chords, i.e. the context, and a probe tone was played. Then people were asked how well the probe tone fitted the context. This resulted in the profiles above.

To get the durations of each of the pitch classes, we use a chromagram representation, which is a time-chroma representation. We decided to only count chroma values, for which the intensity is higher than or equal to 0.8, i.e. we used a threshold of 0.8. We experimented with an implementation with and without the threshold and observed that with this threshold the algorithm performed better.

Given the durations of each of the pitch classes, we calculate the correlation coefficient for all keys by pairing the durations to the profile values. All correlations are calculated by rotating the durations vector (starting with C as key, then C#, etc, ending with B) and calculating the correlation of each rotation (i.e. each key) and the profiles. Note that we now have correlations for each possible major and minor key. Then, the key with the highest correlation is used as the key of the music piece. The correlation is calculated using the formula given below.

$$R = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Formula for calculating the correlation coefficient

In addition to finding the key of a whole music piece, we tried to split a piece using a self-similarity matrix of a chromagram. As a distance measure we used the cosine distance, which is 1 - cosine similarity, given below:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|_2 \|\mathbf{B}\|_2} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Formula for cosine similarity

From this matrix, we computed a novelty curve by sliding a Gaussian checkerboard kernel over the diagonal. At each step the correlation is calculated of the kernel and the underlying block in the matrix. From this we can plot a novelty curve. A peak in this curve corresponds to a possible boundary, and may indicate a key change. Therefore, we split a music piece where these peaks occur and ran the algorithm on these separate pieces, so that we may detect key changes in a music piece.

Results of the algorithm are presented in the last section, where we compare the results with the other algorithms, presented later. We present results of the algorithm run on the whole piece and results run on the pieces splitted by the self-similarity matrix approach.

References

1. Krumhansl, Carol L. *Cognitive foundations of musical pitch*. Oxford University Press, 2001.
2. Hart, R. (2012, August 19). Key-finding algorithm. Retrieved from http://rnhart.net/articles/key-finding/

Key Detection by prefiltering with intensity threshold

In this version of our key detection algorithm we proceeded as follows:

1. Load the audio file that we want to perform our key analysis on.
2. Compute a chroma for every sample
3. For each sample filter out the notes that have an intensity lower than the predefined threshold value T.
4. Merge samples in groups of N seconds by taking the average of the samples intensities.
5. Compute a key matching score per sample group of N seconds as follows:
    a. For each key:
        i. Add the intensities of notes that are in the key to the score
        ii. Subtract the intensities of notes that are not in the key from the score.
    b. The best score determines the the key of that group.

Determining the size of N

Since a single sample often contains a small amount of notes/harmony and only a small part of the complete song it is better to look at the key at a larger time interval. If we would not do this it would be hard, if not impossible, to determine if a Cmajor chord would belong to the Cmajor key or to an Fmajor or Gmajor key. We need to look at a larger interval to gather more information about the harmony of that part of the song. For example a Cmajor chord followed by a Fmajor and Gmajor chord could already make the likelihood that this is a Cmajor key very large.

The downside of making N too large is that it has a lower potential on detecting key changes. If a song does not modulate over time the most accurate way to find the key is to make N the size of the complete song, which results in one key prediction output for the whole song. However, for songs that do modulate we want to make sure that N is not too large. If N would be too large we would mix the modulation of the bridge with other parts of the song which are not transposed and might potentially impact the result.

When comparing the algorithms we computed the results for N = 10 and for N = size of complete song.

Determining the threshold T

The reason why we apply the threshold filter before adding the intensities of the notes is because we want to get rid of noise as soon as possible. When a D note is played, often there are also frequencies leaked to the nearby notes Db and D#. Because these leaked frequencies can often be quite high in intensities we want to filter them out as soon as possible. If we would not filter them out and we would play a D and an E, the average intensity of D#/Eb would get very high compared to the D and E note when averaging the samples and might result in wrong key detection. The threshold that we used while calculating the results was 0.75.

Key Detection by Counting Notes

This key detection method is based on counting the number of occurrences of each note in a piece of music. The number of occurrences are used to give scores to all possible keys in order to find the one that best matches the music. It can be used to detect the key of the full piece or to detect the keys in periods of a chosen number of seconds.

It works as follows:

1. Load the audio file
2. Extract the chromagram of the piece
3. Based on the desired number of seconds, the chromagram is divided into periods of n seconds.
4. For each period the number of occurrences of each note with an intensity higher than 0.9 are listed.
5. This list is used to give a score to each key by adding the counts corresponding to each note that belongs to the key.
6. The key with the highest score is the key that is the best match for that period.

For this algorithm the threshold 0.9 is used to exclude all notes that are not truly played.

Comparing the algorithms

To compare the algorithms we selected songs with different genres. By calculating the key of each of those songs we can see the differences between how each algorithm responses to each genre. In case the prediction is (partly) wrong we will try to provide an explanation on why the algorithm behaves as it does. Because the Krumhansl-Schmuckler algorithm determines music boundaries itself as opposed to the other algorithms, which use a static frame of N seconds.

Classical Music

Song: Brahms: Waltz No. 15 in A-Flat Major, Op. 39
URL: https://www.youtube.com/watch?v=dCAT6sbpCi4
Key: Ab Major

Results:

Algorithm	Whole song	N = 10
Counting notes	Ab-major	2x Ab-maj 1x Eb-maj
Prefiltering with intensity threshold	Ab-major	3x Ab-maj
Krumhansl-Schmuckler with Threshold	Ab-major	-

Jazz Music

Song: Turnabout Jazz Soul - Track 7 - Trials and Tribulations - Court Begins Blue Note Scale
URL: https://www.youtube.com/watch?v=AfNQ-Igw1tY
Key: Blues ladder?

Results:

Algorithm	Whole song	N = 10
Counting notes	C#-major	13x C#-maj 8x Ab-maj 4x F-maj 3x C-maj 1x Bb-maj

Prefiltering with intensity threshold	C#-major	14x C#-maj 7x Ab-maj 5x F-maj 1x Eb-maj 1x C-maj 1x Bb-maj
Krumhansl-Schmuckler with Threshold	F-minor	-

Note: Jazz music does usually not stay in the same key and uses notes outside of the key, this makes it harder for our algorithms to detect a single key.

Piano Music

Song: Twinkle Twinkle
URL: ?
Key: C Major

Results:

Algorithm	Whole song	N = 10
Counting notes	C-major	2x C-maj 2x F-maj 1x 'A-maj
Prefiltering with intensity threshold	C-major	4x C-maj 1x A-maj
Krumhansl-Schmuckler with Threshold	C-major	-

Note: The reason why one group of N=10 was sampled as A-major is because the first 10 seconds of this song are silent (but containing some background noise), which apparently contained frequencies closest to the A-major key.

Pop Music

Song: One Direction - Story of My Life
URL: https://www.youtube.com/watch?v=W-TE_Ys4iwM
Key: Eb major

Results:

Algorithm	Whole song	N = 10
Counting notes	Eb-major	13x Eb-maj 4x Bb-maj 3x Ab-maj 3x C#-maj 1x F-maj
Prefiltering with intensity threshold	Eb-major	12x Eb-maj 5x Ab-maj 5x Bb-maj 2x C#-maj
Krumhansl-Schmuckler with Threshold	Eb-major	-

Metal Music

Song: Metallica - Sad But True
URL: https://www.youtube.com/watch?v=A8MO7fkZc5o
Key: G-major

Results:

Algorithm	Whole song	N = 10
Counting notes	Eb-major	22x Eb-maj 4x Bb-maj 3x C-maj 2x A-maj 1x Ab-maj
Prefiltering with intensity threshold	Eb-major	23x Eb-maj 4x Bb-maj 2x A-maj

		1x G-maj 1x C-maj 1x Ab-maj
Krumhansl-Schmuckler with Threshold	D-major	-

Note: Metal music has a lot of distortion, power-chords (which lack a third) and usually some atonal elements, this makes it harder for the algorithms to detect a single key.

Orchestrated Music

Song: Tchaikovsky -Dance Of The Sugarplum Fairy
URL: https://www.youtube.com/watch?v=Rapf3g_XvCc
Key: G-major

Results:

Algorithm	Whole song	N = 10
Counting notes	G-major	8x G-maj 2x C-maj 2x F-maj 2x E-maj
Prefiltering with intensity threshold	G-major	7x G-maj 3x C-maj 3x F-maj 1x E-maj
Krumhansl-Schmuckler with Threshold	E-minor (G-major)	-

Movie Music

Song: Inception - Time
URL: https://www.youtube.com/watch?v=RxabLA7UQ9k
Key: G-major

Results:

Algorithm	Whole song	N = 10
Counting notes	G-major	6x Eb-maj 6x G-maj 6x F-maj 3x D-maj 3x A-maj 1x C-maj 1x Bb-maj 1x Ab-maj
Prefiltering with intensity threshold	G-major	8x Eb-maj 6x F-maj 5x G-maj 4x D-maj 2x Ab-maj 2x A-maj
Krumhansl-Schmuckler with Threshold	G-major	-

Modulated Music

Song: Beyoncé - Love On Top
URL: https://www.youtube.com/watch?v=Ob7vObnFUJc
Key: C major, Db major, D major, Eb major, E major

Results:

Algorithm	Whole song	N = 10
Counting notes	F-major	5x C#-maj 2x G-maj 2x C-maj 2x Bb-maj 2x E-maj 2x Ab-maj 2x A-maj 1x B-maj

Prefiltering with intensity threshold	F-major	7x C#-maj 3x C-maj 2x Bb-maj 2x E-maj 2x Ab-maj
Krumhansl-Schmuckler with Threshold	C-major	-

Note: Because of the modulations our algorithms were not able to detect a single key for the entire song.

Christmas Music

Song: Dean Martin - Let it Snow
URL: https://www.youtube.com/watch?v=o2uvtl-1V70
Key: C#-major (Verse 3 in Ab-major)

Results:

Algorithm	Whole song	N = 10
Counting notes	Ab-major	5x Ab-maj 2x A-maj 2x C#-maj 2x Eb-maj
Prefiltering with intensity threshold	Ab-major	6x Ab-maj 2x A-maj 2x Eb-maj 1x C#-maj
Krumhansl-Schmuckler with Threshold	Ab-major	-

Note: Since the keys C#-major and Ab-major are so similar, all of our algorithms think that the song is in Ab-major instead of C#-major, probably because the bridge which is written in Ab-major also influences the overall outcome a lot.

Hardstyle

Song: Dragonborn
URL: https://www.youtube.com/watch?v=VaiHTvifGt0
Key: G Major

Results:

Algorithm	Whole song	N = 10
Counting notes	G major	14x G-maj 11x Eb-maj 2x B-maj 1x Ab-maj
Prefiltering with intensity threshold	G major	13x G-maj 8x Eb-maj 4x Ab-maj 3x B-maj
Krumhansl-Schmuckler with Threshold	G major	-

Conclusion

Each of the algorithm is generally good at discovering the classical, pop, piano, orchestrated, movie and hardstyle tracks. Jazz, Metal, Modulated songs (including the Christmas track) are harder to recognize. Jazz is general harder to recognize because the music generally differentiates a lot from the notes belonging to the key. Metal contains a lot of distortion which contain a lot of different frequencies that are not within the scale either. The modulated track, including the christmas track, generate a wrong key for the overall song because the average of all the modulations is not necessarily the key on which overall song is generally based (if you can even talk about overall key).
Adding segmentation using a self-similarity matrix did not give good results. There were too many short segments, which makes it harder for the algorithm to find the correct key. This could also result in a group which has a partial modulation and a partial non-modulation which is likely to lead to faulty key recognition. In the future these algorithms could be applied on more logically based sample groups as well.