Homework 2

Key-finding algorithm

Li Su
Research Center for IT Innovation, Academia, Taiwan
lisu@citi.sinica.edu.tw

(You don't need any solid understanding about the musical key before doing this homework, but we believe that you will learn the musical meaning of key after doing this homework!)

The "key" is one of the most important attribute in our music. Given a musical scale, the key defines the "tonality", namely the *tonic* note and the tonic chord, and the "mode": whether it is a major key or a minor key. Usually (but not always), the tonic note is recognized as "the first note" or "the last note" in a music piece. Moreover, if the chord corresponding to the tonic (i.e. the tonic chord) is a major triad, then the key should be a major key. On the other hand, if the tonic chord is a minor triad, then the key should be a minor key. However, in real-world musical data processing we might not know when the first or the last note appears, because sometimes we are given only a fragment of a song. This is the case we meet for the music dataset (GTZAN) provided in this assignment.

How to identify the tonic note together with the major/minor scales? This is fundamental in music training, but there seems to be no clear and direct method telling us how to do this. In this assignment we will design some key finding algorithm by utilizing the pitch structures in the chroma feature.

(PS: Do not confuse the major/minor key with the major/minor chord. A chord is the co-occurrence of (usually 3) notes, like the major triad and the minor triad, while a key is referred to as the structural information in a diatonic scale.)

Recall that a diatonic scale is shared by two relative keys, one is major and the other is minor. For example, the relative minor key of the C major key is the A minor key. More importantly, recall that the main difference between the major scale and the minor scale is the position of the semitone with respect to the tonic. Denote T as the tone and S the semitone, a major scale is represented as T-T-S-T-T-T-S while a minor scale is T-S-T-T-S-T-T, from the tonic to the leading tone. See the following figures.

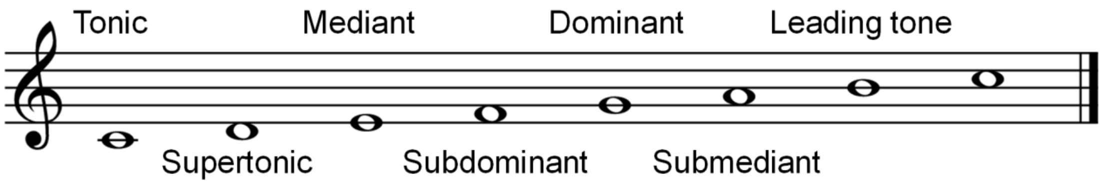

Figure: diatonic scale.

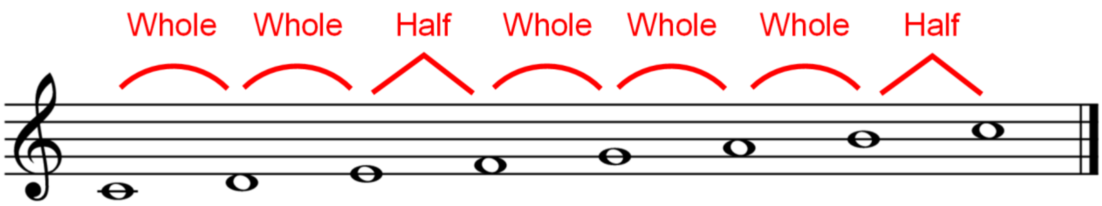

Figure: C major scale.

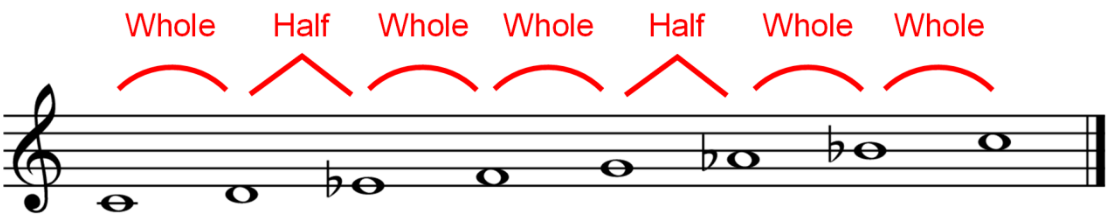

Figure: C minor scale.

Prerequisite:

(1) Download the GTZAN dataset from:
http://marsyas.info/downloads/datasets.html

(2) Download Alexander Lerch's annotation of key on the GTZAN dataset from:
https://github.com/alexanderlerch/gtzan_key

(3) For MATLAB users, download the Chroma Toolbox from:
http://resources.mpi-inf.mpg.de/MIR/chromatoolbox/

(4) For Python users, download librosa and refer to librosa.feature.chroma_stft:
http://bmcfee.github.io/librosa/generated/librosa.feature.chroma_stft.html#librosa.feature.chroma_stft

Method 1: Binary template matching, with tonic note obtained from the term frequency

In this method, we assume that the tonic pitch is the one which *appears most often*. Therefore, a simple idea of finding the tonic pitch is to (1) summing up all the chroma features of the whole music piece into one chroma vector (this process is usually referred to as sum pooling), (2) finding the maximal value in the chroma vector, and (3) considering the note name corresponding to the maximal value as the tonic pitch. Given a chromagram $\mathrm{C} = [\mathrm{c}_1, \mathrm{c}_2, \dots, \mathrm{c}_N]$, $\mathrm{c}_\mathrm{i} \in R^{12}$, where N is the number of frames, the summed chroma vector is

$$x = \sum_{i=1}^{N} c_i$$

Knowing the tonic, the next step is to find the diatonic scale "embedded" in the music piece. Based on the idea of template matching, this can be done by finding the correlation coefficient between the summed chroma features and the template for the diatonic scale. For example, if we have found that the tonic is C, then we generate two templates, one for C major key and the other for c minor key:

$$\mathrm{y}_{\mathrm{C\ Major\ key}} = [1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 1]$$
$$\mathrm{y}_{\mathrm{c\ minor\ key}} = [1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0]$$

While the first index is for C note, the second for C# note, …, and the last index for B note. The correlation coefficient is,

$$R(x,y) = \frac{\sum_{k=1}^{12}(x_k - \bar{x})(y_k - \bar{y})}{\sqrt{\sum_{k=1}^{12}(x_k - \bar{x})^2 \sum_{k=1}^{12}(y_k - \bar{y})^2}}$$

where x is the summed chroma vector and y is the template for a key. There are 24 possible keys, and according to Alexander Lerch's annotation, they are indexed as (upper case means major key and lower case means minor key):

A	A#	B	C	C#	D	D#	E	F	F#	G	G#
0	1	2	3	4	5	6	7	8	9	10	11
a	a#	b	c	c#	d	d#	e	f	f#	g	g#
12	13	14	15	16	17	18	19	20	21	22	23

If the tonic number is given as $0 \le j \le 11$, we only have to compare the correlation coefficients between $R\left(x, y^{(j)}\right)$ (major key) and $R\left(x, y^{(j+12)}\right)$ (minor key). If $R\left(x, y^{(j)}\right) > R\left(x, y^{(j+12)}\right)$, we say the music piece is in the *j* major key, otherwise it is in *j* minor key. A music piece only has one key. The accuracy of key finding can therefore be define as

$$\text{ACC} = \frac{\#\text{ of correct detection}}{\#\text{ of all music pieces}}$$

Q1 (25%): Compute the CLP (chromagram with logarithmic compression) feature using the Chroma Toolbox or implement it in python based on librosa. Set the *factor of logarithmic compression* to be 100:

paramCLP.factorLogCompr=100;

This factor was referred to as γ of $\log(1+\gamma|x|)$ in our course slides page 37 in Lecture 05. Please refer to the demo programs in the Chroma Toolbox for the details. Use the binary template matching idea mentioned above for key finding in all the music pieces in the GTZAN dataset, and compare your estimation with the ground truth annotation of key in the dataset. What's the overall accuracy (ACC)? If we group the music pieces according to genre (i.e. dividing them to "Pop", "Blues", "Metal", "Hip-hop" and "Rock"), what's the key detection accuracy for each genre? Which genres have lower accuracy and can you guess why (from musical point of view)?

Note that some of the music pieces have unknown key labels, so please don't count these pieces while calculating the accuracy.

Q2 (20%): Adjust the factor of logarithmic compression to different values, say, 1, 10, 100, and 1000. Repeat the experiment in Q1 and discuss how this factor is related to the result.

Q3 (25%): You might have found that some of the error detection results behave similarly. For example, C major key is easily to be detected as G major key (a perfect-fifth error), A minor key (a relative-major/minor error), or C minor key (a parallel-major/minor key), because these erroneous keys are intrinsically "closer" to C major keys than others. Therefore, in MIREX key detection competition, these closely related keys are considered in the scoring of key detection:

Relation to correct key	Points
Same	1.0
Perfect fifth	0.5
Relative major/minor	0.3
Parallel major/minor	0.2
Other	0.0

Therefore, the new accuracy is defined by:

$$\text{ACC} = \frac{\#\ \text{Same} + 0.5(\#\ \text{Fifth}) + 0.3(\#\ \text{Relative}) + 0.2\ (\text{Parallel})}{\#\ \text{of all music pieces}}$$

Use this new accuracy to evaluate the experiment in Q1 and discuss the result.

Method 2: Krumhansl-Schmuckler key-finding algorithm

A more advanced set of templates for key detection is the Krumhansl-Schmuckler (K-S) profile. Instead of using a binary (0 or 1) templates as we did before, we assign numerical values to the template according to the profile numbers shown in the following Table (see the columns labeled by K-S). These values came from an experiment of human perception. The experiment is done by playing a set of context tones or chords, then playing a probe tone, and asking a listener to rate how well the probe tone fit with the context.

Therefore, in Method 2, we consider using the correlation coefficient between the input chroma features and the K-S profile for key detection. Notice that the major and minor profiles are rendered by different values. In this task we don't need to probe the tonic first, but just need to find the maximal correlation coefficient among the major profile, minor profile, and the 12 circular shifts of them, respectively. A web resource http://rnhart.net/articles/key-finding/ nicely demonstrates this idea.

Q4 (30%): Use the the Krumhansl-Schmuckler's method to do the same task in Q1, Q2, and Q3 and discuss the experiment result.

Major key			Minor key		
Name	Binary	K-S	Name	Binary	K-S
Tonic	1	6.35	Tonic	1	6.33
	0	2.23		0	2.68
Supertonic	1	3.48	Supertonic	1	3.52
	0	2.33	Mediant	1	5.38
Mediant	1	4.38		0	2.60
Subdominant	1	4.09	Subdominant	1	3.53
	0	2.52		0	2.54
Dominant	1	5.19	Dominant	1	4.75
	0	2.39	Submediant	1	3.98
Submediant	1	3.66		0	2.69
	0	2.29	Leading tone	1	3.34
Leading tone	1	2.88		0	3.17

Bonus Task: What is the limitation of these two methods in key detection? And is there any drawback of using the GTZAN dataset for key detection? For example, do you think 30 seconds is long enough for key detection? Discuss these issues, and, if possible, please design an algorithm that outperforms the two algorithms introduced here in at least two of the five genres considered in this assignment.

The grading policy (e.g. about delay in HW submission) is the same as HW1. Please send your zip file containing the report and your codes, with email title "HW2 [your ID]" to lisu@citi.sinica.edu.tw.

The deadline for this homework is April 18, and we will discuss it on April 21.