Optimization of Voice/Music Detection in Sound Data

Shin'ichi Takeuchi, Masaki Yamashita, Takayuki Uchida, Masahide Sugiyama

Graduate School of Computer Science and Engineering
University of Aizu, Japan
m5051121@u-aizu.ac.jp

Abstract

Automatic voice/music segment detection is expected for various applications. For the general applications of voice recognition and dictation, input voice for the recognition is needed to detect and remove music section automatically.

In order to detect voice and music segments, where sound data contains both voice and music, this paper proposes weighted Block Cepstrum Flux (BCF) and optimizes the weight vector using discriminative training technique.

This paper also discusses the effectiveness of the frequency axis weighting in calculating Cepstrum Flux and BCF. Here, frequency axis weighting is carried out by the modification of LPC Cepstrum distance calculation.

The experimental results shows the detection error rate of the original BCF is 11.56% and the error rate of the weighted BCF with the low-frequency weighting for closed data is 9.08 %, and 10.48 % for open data. This rensult shows the effectiveness of both time and frequency axis weighting in BCF calculation for detection between voice and music.

1. Introduction

Several feature parameters were proposed to separate voice and music. This paper describes the optimization of feature parameters based on a discriminative training technique.

Spectral Flux has been proposed by [1] and showed its potential with an experiment for segmenting voice and music. Cepstrum Flux and Block Cepstrum Flux has been proposed by [2, 3] as natural extensions of Spectral Flux and showed its effectiveness. Sound data used for the experiment includes sound and voice, in alternation. The effectiveness for sound data in real world has been evaluated the performance using actual data [4, 5]. Now, research is developing to detecting music [6] and voice [7].

Originally, Spectral Flux parameter has been proposed and showed its potential with an experiment for segmenting voice and music and Cepstrum Flux has been introduced as an extension of Spectral Flux in the time axis. The fundamental idea is to characterize sound data by comparing the base frame to the past frames in Cepstrum vector of sound data. Block Cepstrum Flux (BCF) has been introduced to improve Cepstrum Flux and the results of the preliminary experiment showed its effectiveness in voice/music segment detection. Block Cepstrum Flux is calculated by averaging Cepstrum Flux in the definite time block.

This paper is organized as follows: The next sections describes the preparations. Section 3 describes the discriminative training technique and optimization of the error rate. Sections 4 describes experiments, and finally section 5 presents the conclusion of this paper.

2. Preparations

2.1. Cepstrum Flux & Block Cepstrum Flux

Cepstrum Flux is an extension of Spectral Flux toward the time axis. It is a method to characterize sound data by comparing plural LPC Cepstrum vectors to based frame in sound data alternately. Cepstrum Flux is defined by Eq.(1).

$$D_n(J) = \frac{1}{J}\sum_{j=1}^{J} \|\boldsymbol{c}_n - \boldsymbol{c}_{n-j}\|^2 . \tag{1}$$

J is the size of the time window (number of frames) and $\boldsymbol{c}_n$ is a LPC Cepstrum vector at time n. The theoretical properties for Cepstrum Flux is described in [8].

Block Cepstrum Flux is acquired by averaging Cepstrum Flux at regularized time blocks. Block Cepstrum Flux is defined by Eq.(2).

$$B_n(W) = \frac{1}{W}\sum_{i=0}^{W-1} D_{n-i}(J). \tag{2}$$

W is the size of the time window (number of frame per 1 block) and D_n is a value of Cepstrum Flux at time n. If the value of B_n is smaller than threshold, the frame n is a voice, and if the value of B_n is greater than threshold, the frame n is a music.

2.2. Detection error rate

The distinction between voice and music is determined by the value of B_n and its threshold T, so the detection error rate is calculated as follows:

$$\phi(B_n(W) - T) = \begin{cases} 1 & B_n(W) > T \quad \text{voice} \\ 0 & B_n(W) < T \quad \text{music}. \end{cases} \tag{3}$$

Here, $\phi(x)$ is a step function.

$$\phi(x) = \begin{cases} 1 & x \geq 0 \\ 0 & \text{otherwise}. \end{cases} \tag{4}$$

The teaching signal that indicates voice/music at time n is defined as follows:

$$t_n = \begin{cases} 1 & n\text{: voice} \\ 0 & n\text{: music}. \end{cases} \tag{5}$$

The detection error rate E ($0 \leq E \leq 1$) is calculated by Eq.(6).

$$E = \frac{1}{N}\sum_{n=1}^{N}(\phi(B_n(W) - T) - t_n)^2. \tag{6}$$

Here, N is the number of frames.

2.3. Optimization using a gradient method

Step function $\phi(x)$ described at Eq.(6) is discontinuous at $x = 0$, so it is replaced with following monotonous increasing sigmoid function to give differentiability.

$$\phi_a(x) = \frac{1}{1 + e^{-ax}} \quad (a > 0).$$

$\phi_a(x)$ is $0 < \phi_a(x) < 1, \phi_a(0) = 1/2$, and gets closer to the step function when a increases. The derivative of $\phi_a(x)$ is calculated using $\phi_a(x)$:

$$\phi'_a(x) = a\ \phi_a(x)\ (1 - \phi_a(x)).$$

$\phi'_a(x) = \phi'_a(-x)$, and $0 < \phi'_a(x) \leq a/4$. The error rate can be replaced in the following Eq.(7) using the sigmoid function.

$$E_a(\boldsymbol{w}, T) = \frac{1}{N}\sum_{n=1}^{N}(\phi_a(B_n(W) - T) - t_n)^2. \quad (7)$$

E_a is an approximation of the detection error rate E and $\lim_{a \to +\infty} E_a = E$. The partial derivatives of w_k and T which are the variables of the criterion function E_a are calculated for optimization using a gradient method.

$$\frac{\partial E_a}{\partial w_k} = \frac{2}{N}\sum_{n=1}^{N}(\phi_a(B(W) - T) - t_n) \times \phi'_a(B(W) - T)\frac{\partial B_n(W)}{\partial w_k}, \quad (8)$$

$$\frac{\partial E_a}{\partial T} = -\frac{2}{N}\sum_{n=1}^{N}(\phi_a(B(W) - T) - t_n) \times \phi'_a(B(W) - T). \quad (9)$$

The weight parameters are updated by the following Eq.(10) when the number of iterations is t.

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \varepsilon\ \nabla E_a(\boldsymbol{w}_t). \quad (10)$$

Here, the following equations are introduced.

$$\nabla E_a = \left(\frac{\partial E_a}{\partial w_k}\right)\ (k = 0, 1, \ldots, W).$$

3. Optimization of Block Cepstrum Flux using discriminative training technique

Block Cepstrum Flux is calculated about frequency axis and time axis, so it can be weighted about the frequency and time axis. As methods to give the weight parameters about time axis, the first one is to give a weight parameter for Cepstrum Flux of Eq.(1), and second one is for Block Cepstrum Flux of Eq.(2). On the other hand, the method to give weighting on frequency axis is to weight on LPC Cepstrum distance.

In this paper, the first method to weight about time axis is called Block Weighted Cepstrum Flux (BWCF), and second one is called Weighted Block Cepstrum Flux (WBCF).

3.1. Weight parameters on time axis

Block Weighted Cepstrum Flux (BWCF)

Cepstrum Flux in Eq.(1) is weighted as shown in Eq.(11).

$$D_n(J) = \frac{1}{J}\sum_{j=1}^{J} w_j \|\boldsymbol{c}_n - \boldsymbol{c}_{n-j}\|^2. \quad (11)$$

w_j is a weight parameter that is given to $d_n^j = \|\boldsymbol{c}_n - \boldsymbol{c}_{n-j}\|^2$ calculating Cepstrum Flux at time n. Therefore, E is a $J+1$ variable function that has a variable $\boldsymbol{w} = (w_1, \cdots, w_J, T)$. Thus, Eq.(8) is calculated as follows:

$$\frac{\partial B_n(W)}{\partial w_k} = \begin{cases} \frac{1}{WJ}\sum_{i=0}^{W-1} d_{n-i}^k & (1 \leq k \leq J) \\ -1 & (k = J+1). \end{cases} \quad (12)$$

$$\frac{\partial E_a}{\partial w_k} = \begin{cases} \frac{2}{NWJ}\sum_{n=1}^{N}(\phi_a(B_n(W) - T) - t_n) \\ \times \phi'_a(B_n(W) - T)\sum_{i=0}^{W-1} d_{n-i}^k, \\ (1 \leq k \leq J) \\ -\frac{2}{N}\sum_{n=1}^{N}(\phi_a(B_n(W) - T) - t_n) \\ \times \phi'_a(B_n(W) - T)\ \ (k = J+1). \end{cases} \quad (13)$$

Weighted Block Cepstrum Flux (WBCF)

Block Cepstrum Flux in Eq.(2) is weighted as shown in Eq.(14).

$$B_n(W) = \frac{1}{W}\sum_{i=0}^{W-1} w_i D_{n-i}(J). \quad (14)$$

w_i is a weight parameter given to $D_{n-i}(J)$ calculating Block Cepstrum Flux at time n. Therefore, E is a $W+1$ variable function that has a variable $\boldsymbol{w} = (w_0, \cdots, w_{W-1}, T)$. Thus, Eq.(8) is calculated as follows:

$$\frac{\partial B_n(W)}{\partial w_k} = \begin{cases} \frac{D_{n-k}}{W} & (0 \leq k \leq W-1) \\ -1 & (k = W). \end{cases} \quad (15)$$

$$\frac{\partial E_a}{\partial w_k} = \begin{cases} \frac{2}{NW}\sum_{n=1}^{N}(\phi_a(B(W) - T) - t_n) \\ \times \phi'_a(B(W) - T)\ D_{n-k}, \\ -\frac{2}{N}\sum_{n=1}^{N}(\phi_a(B(W) - T) - t_n) \\ \times \phi'_a(B(W) - T). \end{cases} \quad (16)$$

3.2. Weight parameters on frequency axis

The method to add weight parameter about LPC Cepstrum distance has been proposed and showed its effectiveness by [10].

The LPC Cepstrum distance of Cepstrum Flux in Eq.(1) is weighted about time axis. For convenience, 1st order differential filter $(1 - \alpha z^{-1})$ is used as weight parameter function. The LPC Cepstrum distance in Eq.(1) becomes as follows:

$$d^2_{\mathrm{FWCEP}}(f, g) = 2\{\sum_{s=1}^{\infty} d_s^2 + w_1 \hat{e}_1 - 2(w_1 d_1)^2\} \quad (17)$$

$$= 2\{\sum_{s=1}^{M} d_s^2 + 2w_1\sum_{s=1}^{M} d_s d_{s+1} - 2(w_1 d_1)^2\}.$$

For n-th cepstrum coefficients $c_n^{(f)}$ and $c_n^{(g)}$ of LPC spectra $f(\lambda)$ and $g(\lambda)$, d_n and $\hat{e}_n$ are defined as follows:

$$d_n = c_n^{(f)} - c_n^{(g)},\ \hat{e}_n = \sum_{\substack{s-t=n \\ s\neq 0, t\neq 0}} d_s d_t.$$

Here, the weight parameter is

$$w_1 = -\frac{\alpha}{1 + \alpha^2}.$$

When $\alpha < 0$, it means to add weight parameters to low frequency, and when $\alpha > 0$, it means to add weight parameters to high frequency. M is the trancation order and is equal to 16.

3.3. Reduction of computation complexity

The term $\sum_{i=0}^{W-1} d_{n-i}^k$ at BWCF in Eq.(12) is independent with weight parameters and is used at repeated by sum calculation. Therefore d_n^k, s_n^k is calculated in advance for efficiency and recurrence relation is calculated for reducing sum calculation at iterative calculation shown in Eq.(18). For this reduction, the amount of calculation reduce from jnw to $(2j + w - 2)n$. The result of this effectiveness is described at 4.5.

$$\begin{cases} s_n^k = \sum_{i=0}^{W-1} d_{n-i}^k \\ s_n^k = s_{n-1}^k + (d_n^k - d_{n-W}^k) \ \ (k = 1, 2, \ldots, J) \\ B_n(W) = \frac{1}{WJ} \sum_{j=1}^{J} w_j s_n^j. \end{cases} \quad (18)$$

4. Evaluation experiments to detect voice/music section

4.1. Database for evaluation and sound analysis conditions

The experiments use CampusWave database [9] which includes voice and music. CampasWave database consists of 15 sets of FM radio program (15 weeks) and the length of each data set is one hour. They are composed of requested musics, conversation between two female personalities, and commercials. They were recorded at a studio and converted into a computer through DAT link at its sampling frequency of 44.1kHz. The chosen sampling frequency of the database was 16kHz because it includes music. The quantization method is 16bit PCM. They are made in stereo since music is usually recorded in stereo, but mixed to create a single channel for analysis. Table 1 shows the ratio of music, voice, etc. Each set of data has a sound label (e.g., music, voice, silence, ...). The teaching signal t_n in Eq.(5) is generated based on this label information. The experiments use first 10 sets of data (CW01-CW10). The voice section and music section have almost the same ratio: 40% of data. The "others" in Table 1 are BGM and CM.

Table 1: *Database structure (sec.)*

data ID	music	voice	silence	others
CW01	1187	1612	70	667
CW02	1400	1474	72	713
CW03	1442	1603	124	523
CW04	1322	1704	94	524
CW05	1596	1294	96	645
CW06	1435	1603	116	515
CW07	1357	1696	94	508
CW08	1310	1611	125	616
CW09	1467	1450	93	609
CW10	1555	1386	100	537
average	1407	1543	98	585
	(38.7%)	(42.5%)	(2.7%)	(16.1%)

Table 2 shows sound analysis conditions. The initial value of threshold T is set that E is the original error rate of Block Cepstrum Flux. The number of all frames is 229,215 and voice/music frames is $N = 180,011$.

4.2. Evaluation of weighting on time axis

The error rate E_a (Eq.(7)) decreases monotony. Figure 1 shows decrease of error rate E (Eq.(6)). Experiment conditions is as follows: $J = 5, 10, 15$, $W = 400, a = 10, \epsilon = 0.1$. The threshold T does not change because changing of weight parameters has same effects.

Table 2: *Sound analysis conditions*

Sampling Rate	16kHz
Window function	hamming window
Window Length	256 points (16ms)
analysis update cycle	256 points (16ms)
LPC analysis	14 dimensions
Cepstrum analysis	16 dimensions
sound feature value	LPC Cepstrum vector
Window of Cepstrum Flux	$J = 5, 10, 15$
Window of Block Cepstrum Flux	$W = 63, 400$
sigmoid parameter	$a = 10$
iteration parameter	$\epsilon = 0.1 - 0.4$
number of iterations	100

By the comparison of BWCF and WBCF, BWCF gives lower error rate than WBCF when $J = 10, 15$, but BWCF decreases monotony and takes the lowest error rate when $J = 5$. Therefore, the following experiments use BWCF.

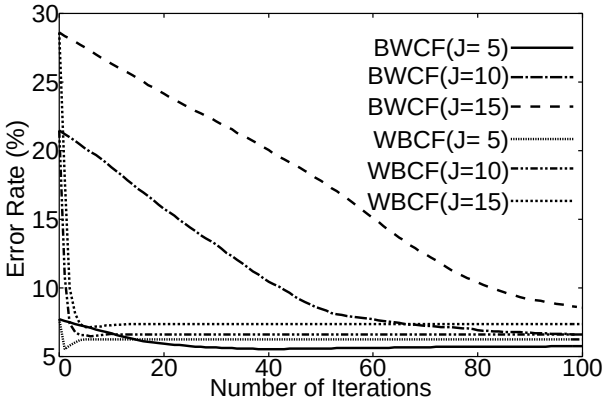

Figure 1: *Comparison of decreasing error rates for BWCF and WBCF*

Table 3 shows initial error rates of 10 data sets and average one after training. The error rate can be reduced by the discriminative training technique. Figure 2 shows envelope of weight parameters.

Table 3: *Average of error rates for 10 data sets(%)*

	with training	
initial error rate	BWCF	WBCF
11.56	10.78	11.04

$J = 5, W = 400$, number of iteration: 100

4.3. Evaluation using open data

Table 4 shows a result of experiment using time axis weight parameters given in 4.2. For open data, the result shows it can not be less than initial error rate 11.56%.

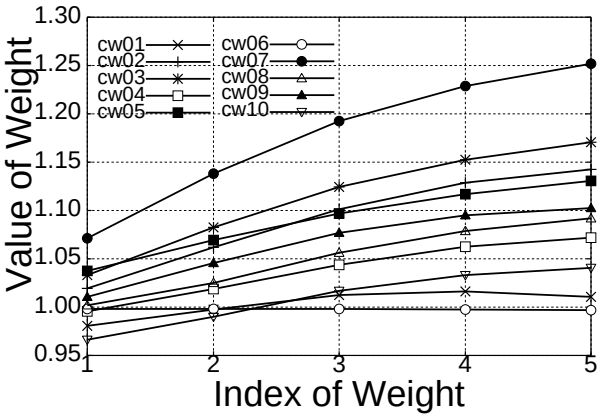

Figure 2: *Shapes of weight parameters*

Table 4: *Error rates using various weight parameters(%)*

weight parameter	close data	open data
initial error rate	11.56	11.56
after 20 iterations	10.74	11.47
after 100 iterations	10.78	12.18
give minimum E	10.48	11.87
give minimum E -5	10.50	11.83

$J = 5, W = 400$, number of iterations is100

4.4. Evaluation of weighting on frequency axis

Figure 3 shows change of error rates when parameter of linear filter α is changed, and Table 5 shows experiment result with frequency axis weighting. Number of iteration is 100.

For this result, when $\alpha = -0.8$, the error rate becomes smallest. The error rate reduces by about 1.1% and becomes 9.16%. This means that to add weight parameters to low frequency is effective.

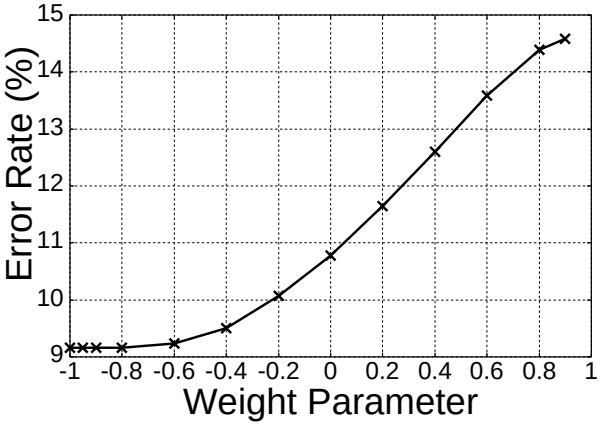

Figure 3: *Relation between frequency weight parameter and error rate*

4.5. Effectiveness for reduction of computation complexity

Table 6 shows the result of experiments for reduction of computation complexity by 3.3. The experiment uses CW02 and the number of iteration is 100. The time to calculate during iteration becomes shorter.

Table 5: *Error rate using frequency axis weighting(%)*

weight parameter	close data	open data
initial error rate	11.49	11.49
after 100 iterations	9.16	10.58
give minimum E	9.08	10.48

$J = 5, W = 400, \alpha = -0.8$

Table 6: *Effectiveness for reduction of computation complexity*

	without	with	ratio
amount of calc.	JNW	$\{W + 2(N-1)\}J$	
number of calc.	360,022,000	1,801,990	199.80
timf of calc.	552.00(s)	93.14(s)	5.92

CPU：Pentium 1GHz
$J = 5, W = 400, N = 180,011$

5. Conclusion

This paper proposes to decrease detecting error rate between voice and music using Cepstrum Flux and Block Cepstrum Flux by using discriminative training technique. As a result, this paper indicates effectiveness of frequency axis weighting in calculating LPC Cepstrum distance, and decrease error rate from 11.56% to 10.48%.

6. References

[1] E.Scheirer, M.Slaney, Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator, Proc. of ICASSP97 (1997).

[2] T.Asano, M.Sugiyama, Segmentation and Classification of Auditory Scenes in Time Domain, Proc. of IWHIT98, pp.13-18 (1998-11).

[3] T.Asano, Study on Auditory Scene Segmentation, MT98-5011101 (1999-03).

[4] T.asano, M.Yamashita, M.Sugiyama, Detection of music section using Cepstrum Flux, ASJ99 3-Q-2, pp.121-122 (1999-09). (in Japanese)

[5] T.Uchida, M.Yamashita, M.Sugiyama, Voice/Music Segmentation using Cepstrum Flux, SP2001-17 (2000-06). (in Japanese)

[6] T.Uchida, Music Segmentation for Sound Streams, MT00-5031105 (2001-03).

[7] M.Yamashita, Automatic Speaker Indexing, MT00-5031124 (2001-03).

[8] M.Sugiyama, Segmentation and Searching of Speaker Segments, Proc. of IWHIT99, pp.15-19 (1999-10).

[9] T.Uchida, M.Sugiyama, Construction of CampusWave Sound Database, ECEI2000, 2A-6 (2000-08). (in Japanese)

[10] M.Sugiyama, K.Shikano, Frequency Weighted LPC Spectral Matching Measures, IECE Trans., J65-A, No. 9, pp.965-972 (1982-09). (in Japanese)